

PROGRAMA de Seminarios

Esta materia no tiene contenidos fijos, sino que puede ser uno de los tres seminarios que aparecen a continuación

Carreras: Tecnicatura Universitaria en Programación Informática - Licenciatura en Informática

Asignatura: Seminarios – Taller de Servicios Cloud

Núcleo al que pertenece: Orientación

Profesores: Arturo Zambrano y Lautaro Woites

Asignaturas Correlativas: No posee

Objetivos:

- Que los alumnos y las alumnas comprendan el paradigma de programación asincrónica
- Que los alumnos y las alumnas sean capaces de identificar situaciones en las cuales aplicar dicho paradigma es beneficioso para el proceso de desarrollo de software.
- Que los alumnos y las alumnas comprendan los beneficios y formas de implementación de APIs RESTful.
- Que los alumnos y las alumnas comprendan los beneficios y mecanismo de implementación de arquitecturas de microservicios.

Contenidos mínimos:

- Paradigma de programación asincrónica usando Node JS ECMAScript 6
- Implementación de APIs RESTful para servicios.
- Deployment de los servicios desarrollados utilizando containers (Docker)

Carga horaria semanal: 4 hs

Programa analítico:

- **Unidad 1: ECMA Script6**
Datatypes, functions, objects, Object Oriented programming in ECMAScript 6.
- **Unidad 2: Asincronismo.**

Definición, implementación mediante callbacks, promesas y nuevas construcciones sintácticas en JS.

- **Unidad 3: RESTful API.**

Invocación de servicios REST, consumo de servicios desde Node JS. Publicación de servicios propios -implementados en Node - como interfaces REST.

- **Unidad 4: Microservicios.**

Docker, administración de containers, deployment de servicios en containers.

Bibliografía obligatoria:

- Zakas, Nicholas. Understanding ECMAScript 6: The Definitive Guide for JavaScript Developers. No Starch Press,US. 2016
- Node.js Notes for Professionals. <https://books.goalkicker.com/NodeJSBook/>
- Leonard Richardson, Sam Ruby, Mike Amundsen. RESTful Web APIs.O'Reilly Media, 2013

Organización de las clases:

Clases teórico prácticas, dos horas teoría y dos de práctica.

Trabajos prácticos que se resuelven parcialmente en el aula y el resto fuera de ella.

Trabajos Prácticos

La práctica de la materia está organizada en diferentes unidades. En sincronía con las prácticas los alumnos y las alumnas deberán implementar un sistema donde aplicarán los conceptos de la lotería y las prácticas.

A continuación describimos el contenido de cada una de las prácticas

Práctica 1: **Introducción a Javascript.** Objetivo: familiarizarse con los conceptos básicos del lenguaje. Expresiones del lenguaje. Tipos de datos básicos. Funciones como elementos de primer orden

Práctica 2: **Javascript y tipos de datos complejos.** Objetivo: Familiarizarse con Arrays y su protocolo. Objetos JS y su protocolo. Resolución de problemas utilizando arrays y objetos JS.

Práctica 3: **Programación orientada a objetos en Javascript.** Objetivo: Comprender y aplicar los conceptos fundamentales de la POO considerando el soporte que brinda ECMAScript6 para ello. Clases, instancias, constructores, metodos. Herencia. Excepciones: lanzamiento de excepciones, manejo de excepciones.

Práctica 4: **JSON**. Objetivo: Comprender qué es y para que se utiliza el formato JSON. Utilización de las herramientas que brinda el lenguaje para trabajar con JSON. Utilizar módulos para poder separar un programa en archivos

Práctica 5: **Asincronismo I**. Objetivo: comprender las diferentes formas de escribir código asincrónico: Callback, promesas. Funciones asincrónicas en Node: I/O archivo y red.

Práctica 6: **Asincronismo II** Formas alternativas de escribir código asincrónico en JS: async/await.

Práctica 7: **REST APIs**. Objetivo: Ejercitar el consumo de APIs REST y las herramientas necesarias para explorarlas. Implementar un servicio rest con express.

Práctica 8: **Containers**. Introducir al alumno y a la alumna al uso de Docker, creación de imágenes, creación de containers. Definición de volúmenes, mapping de puertos. Deployment de servicios Node en un container docker.

El **trabajo práctico** mencionado anteriormente se desarrolla a lo largo de 3 entregas incrementales. A continuación detallamos el alcance y el objetivo de cada una.

Etapas 1: Desarrollo orientado a objetos del core de un sistema escrito en node. La primera versión del sistema se opera mediante una interfaz de línea de comando.

Etapas 2: Sistema desarrollado en la etapa 1 es extendido permitiendo consultar e integrar datos de internet mediante interfaces REST.

Etapas 3: Sistema desarrollado en la etapa 3 es extendido agregando más servicios. Los servicios son desplegados en container que se comunican a través de interfaces REST.

Modalidad de evaluación:

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18).

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.

CRONOGRAMA TENTATIVO

Semana	Teoría	Práctica
1	Presentación de la Materia Importancia de la materia, contexto en el desarrollo actual de	Encuesta, relevamiento de conocimiento de los alumnos y las alumnas.

	servicios. Definiciones básicas, servicios microservicios, programación asincrónica, REST. Historia de JS. Versiones.	
2	Intro Javascript: elementos básicos del lenguaje, tipos básicos, array, object, funciones, funciones de alto orden, conversiones entre tipos, conversiones automáticas	Practica 1
3	Intro Javascript: JSON, Strings, conversiones, cosas raras, funciones aritméticas, callbacks? Clases y objetos	Práctica 2 y 3 Objetos
4	Módulos, Excepciones, JSon	Práctica 4 / Enunciado TP Etapa1
5	Node + event loop , Concurrencia, sync vs async. Callback y Promesas	Práctica 5 Asincronismo I
6	Asincronismo y REST	Práctica 6 Asincronismo II
7	REST	Practica 7 API REST
8	Patrones API Rest	Enunciado TP Etapa2
9	Buffer / Microservicios / Visado	
10	Docker	P8 Docker / Microservicios
11	Micro Servicios clase de ejemplos de sistema conocidos Patrones en Micro Servicios Arquitectura	Enunciado TP Etapa 3
12	Consulta TRABAJO FINAL	
13	Entrega Inicial	
14	Consulta	
15	Reentrega Final	

PROGRAMA de Seminario: Televisión Digital

Carreras: Tecnicatura Universitaria en Programación Informática - Licenciatura en Informática

Asignatura: Seminario: Televisión Digital

Núcleo al que pertenece: Complementario

Profesor: Leonardo Gassman

Asignaturas Correlativas: Estructuras de Datos - Programación con objetos 2

Objetivos:

Que el alumno y la alumna:

- Comprenda las diferencias entre TV analógica y digital
- Conozca los parámetros que afectan la calidad del video y el audio digital
- Conozca la estructura básica de un transport stream
- Conozca las características distintivas de la norma ISDB-Tb
- Comprenda cómo la interactividad cambia la experiencia del usuario y sea capaz de desarrollar contenidos interactivos para TV Digital.

Contenidos mínimos:

Características generales de TV Digital. Audio y video digital. Datos en la TV digital. TV digital interactiva:Ginga. Transport Stream MPEG2. Lenguajes NCL y LUA para el desarrollo de aplicaciones interactivas.

Carga horaria semanal: 4 hs

Programa analítico:

Unidad 1: Introducción

- TV Digital vs Analógica.
- Normas de TV Digital a nivel mundial y regional.

Unidad 2: Video y Audio en la TV Digital

- Sistemas y resoluciones
- 480i y 480p, 576p y 576i, 720p y 720i, 1080i
- Codificación de audio y video
- Calidad: codecs, bitrate

Unidad 3: Transport Stream

- Flujo de Transport (Transport Stream)

- Packetized Elementary Stream (PES)
- Rudimentos de MPEG2 transport stream
- Estructura: PIDs de audio, video datos
- Tablas:
- PAT Program Association Table
- PMT's Program Map Table
- CAT Conditional Access Table
- NIT Network Information Table
- DSM-CC: Carousel de datos y de objetos

Unidad 4: Interactividad

- Middlewares para interactividad: Ginga
- Interactividad Local
- Interactividad con Canal de Retorno
- Stream Events
- Diseño de aplicaciones para TV
- Usabilidad en aplicaciones para TV

Unidad 5: TV Digital en Argentina

- Norma ISDB-Tb
- Layers
- Modulación
- Codecs para audio y video
- Utilización de las frecuencias.

Bibliografía obligatoria:

- Soares, L.F.G. y Barbosa, S.D.J. Programando en NCL- Desenvolvimento de Aplicações para Middleware Ginga, TV figital e Web. Campus, 2009
- Morris S. and Smith-Chaigneau A. Interactive TV Standards: A Guide to MHP, OCAP, and JavaTV. Focal Press, 2005
- Fischer, W. Digital Video and Audio Broadcasting Technology: A Practical Engineering Guide 3rd. Springer Publishing Company, Incorporated, 2010

Bibliografía de consulta:

- Watkinson, J. The MPEG Handbook Focal Press, Woburn (MA), USA, 2001
- Jack, K. Video demystified: A handbook for the digital engineer LLH Technology Publishing, 2001

Organización de las clases:

El dictado de la materia incluye:

- Clases teóricas/prácticas 1 vez por semana
- Trabajos prácticos que abordan los diferentes temas.
- Para la resolución de los trabajos prácticos se utilizarán herramientas basadas en software libre
- Se considera la posibilidad de realizar algunas prácticas sobre hardware especializado que se encuentre disponible

Trabajos Prácticos

Práctica 1: Introducción a la TVD.

¿Qué es la TVD? Características y funcionalidades. Comparación con la TV analógica

Práctica 2: Aplicaciones de TVD en Argentina – Audio y video/ Unidad 2 y 5

Diferentes Estándares y normas, ATSC, DVB, ISDB. Cuál usamos en Argentina.

Práctica 3: Programación bajo nivel (Streams) / Unidad 3

Definición de Streams de bytes. Generación de streams con Python. Generación de streams en tiempo real (constraints).

Práctica 4: Transport Stream: PAT / Unidad 3

Estructura básica de un Transport stream (Tablas, PIDs). Streams dentro del TS (ES, PES). ¿Qué es la PAT? Introducción a OpenCaster, Generación de una PAT con python/opencaster.

Práctica 5: Transport Stream: PMT / Unidad 3

Definición de una PMT. Que puede contener un Programa (Datos, Audio, video). Generación de una PMT programáticamente.

Práctica 6: Transport Stream: Otras tablas / Unidad 3

Otras tablas utilizadas en el transport stream. NIT, SDT, EIT, TDT/TOT. Generación y Relación con el resto de las tablas.

Práctica 7: Interactividad: Ginga/NCL / Unidad 4

Aplicaciones Ginga, ¿Que son? que nos permiten hacer. Estructura de un documento NCL. Desarrollo de una app sencilla.

Práctica 8: Interactividad: Ginga/LUA / Unidad 4

Lenguaje de scripting Lua. Agregado de scripts Lua a aplicaciones NCL. Implementación de una aplicación sencilla NCL/Lua

Práctica 9: Interactividad: Ginga, Stream Events / Unidad 4

Definición de Stream Events, como mecanismo para notificar de eventos “arbitrarios” a los programas. Generación y envío de un Stream Event. Aplicación NCL/Lua sencilla que reacciona ante un stream event.

Trabajo final

El TP final consiste en la implementación de un decodificador simplificado de transport stream de TV Digital. A partir de muestras reales del transport stream, el alumno y la alumna debe implementar la lectura de paquetes y partir de allí encontrar los diferentes streams (audio, video, aplicaciones, etc). El trabajo permite poner en práctica el conocimiento adquirido sobre las diferentes estructuras de tablas del estándar ISDB-T. Finalmente, los streams encontrados se puede decodificar software existente (decodificadores de audio y video, como VLC) para corroborar que han sido identificados exitosamente.

Modalidad de evaluación:

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18).

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.

CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Introducción a la TVD / Unidad 1 y 5	x				
2	Aplicaciones de TVD en Argentina – Audio y video/ Unidad 2 y 5	x				
3	Programación bajo nivel (Streams) / Unidad 3	x		x		
4	Transport Stream: PAT / Unidad 3	x		x		
5	Transport Stream: PMT / Unidad 3	x		x		
6	Transport Stream: Otras tablas / Unidad 3	x		x	x	
7	Interactividad: Ginga/NCL / Unidad 4	x		x		
8	Interactividad: Ginga/LUA / Unidad 4	x		x		
9	Interactividad: Ginga, Stream Events / Unidad 4	x		x	x	
10	Taller trabajo práctico final			x		
11	Taller trabajo práctico final			x		
12	Taller trabajo práctico final			x		
13	Taller trabajo práctico final			x		
14	Exposición trabajo final / Examen Parcial				x	
15	Recuperatorio Examen Parcial				x	
16	Integrador				x	

*INDIQUE CON UNA CRUZ LA MODALIDAD

PROGRAMA de Seminarios

Carrera/s: Tecnicatura Universitaria en Programación Informática / Licenciatura en Informática

Asignatura: Seminario: Programación de Microcontroladores con Tecnologías Libres

Núcleo al que pertenece: Complementario

Profesor: José Luis Di Biase

Prerrequisitos: No posee

Objetivos:

Que la persona:

- Conozca los requerimientos del entorno operativo necesarios tanto para desarrollar como para integrar soluciones de hardware.
- Pueda instalar, configurar, operar y desarrollar aplicaciones que incorporen hardware.
- Integre los conocimientos adquiridos en las otras materias del área para pensar proyectos que incorporen el mundo físico.

Contenidos mínimos:

Se trata de cursos sobre temáticas específicas correspondientes a las características dinámicas del ámbito de la programación, relacionadas con:

- temas avanzados de programación.
- dominios o tipos específicos de proyectos de software.
- herramientas que cuenten con un real interés para la complementación de la formación de quienes cursan la materia.

Carga horaria semanal: 4 hs

Programa analítico:

- Conocer mínimamente conceptos y componentes electrónicos para desarrollar proyectos que incorporen hardware.
- Instalar, configurar y operar software necesario para el proceso de grabación e instalación del firmware en los microcontroladores.
- Conocer y utilizar herramientas básicas que facilitan el desarrollo de firmware para microcontroladores.
- Explorar, investigar y desarrollar formas de conexión y comunicación entre dispositivos de hardware libre.

Bibliografía obligatoria:

- Arduino Cookbook, Michael Margolis, O'Reilly
- Getting Started with Arduino, Massimo Banzi, O'Reilly Media/Make

Bibliografía de consulta:

Libros:

- Beginning Arduino Programming - Brian Evans - Apress
- Practical Arduino - Cool Projects for open source hardware

Sitios web:

- <https://www.adafruit.com/tutorials>
- <http://learn.sparkfun.com/>

Organización de las clases:

Las actividades de enseñanza se desarrollarán con trabajos de laboratorio continuos en los cuales se pondrán en juego los conceptos teóricos necesarios para su concreción in situ.

Introducción a la electrónica

Conocimientos mínimos de electrónica , presentación de diversos componentes y su utilización habitual en casos simples.

Experimentación con Microcontroladores

Comprensión mediante la ejercitación paso a paso del circuitos básico de uso, programación, compilación y grabación de microcontroladores.

Proyectos con Arduino

Experimentación con ejercicios iniciales a reproducir integrando componentes básicos tales como leds, resistencias, servos, potenciómetros, foto resistencias y botones.

Proyectos avanzados con Arduino
Experimentación con ejercicios que incrementan las dificultad incorporando circuitos integrados, sensores, y relays.

Debug de aplicaciones

Utilización de herramientas y estrategias para debug de aplicaciones a bajo nivel.

Controlando Hardware desde PC

Ejercicios prácticos evaluando diferentes modalidades de comunicación entre pc y diferentes tipos hardware incluyendo serie, firmata, websockets y mqtt.

Modalidad de evaluación:

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18).

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.

CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Introducción a la electrónica			x		
2	Microcontroladores y nuevo			x		

	hardware					
3	Primeros proyectos con Arduino			x		
9	Proyectos avanzados con Arduino			x		
16	Debug de aplicaciones			x		
17	Controlando Hardware desde PC			x		
	Trabajo Práctico Integrador de los temas vistos					x

Quilmes, 1 de diciembre de 2025

VISTO: el Estatuto de la Universidad Nacional de Quilmes, y la Resoluciones (CD) CyT N°228/15, (CS) 171/22, (CD CyT) N° 46/25 y

CONSIDERANDO:

Que de acuerdo con el artículo 43° del Estatuto de la Universidad Nacional de Quilmes “Las carreras y las diplomaturas son unidades curriculares y dependen del Departamento responsable del desarrollo de las asignaturas que constituyen el núcleo básico de los respectivos planes de estudio”.

Que el artículo 79° inc. k) del mismo Estatuto establece que los Consejos Departamentales “Aprueban y supervisan los programas de las asignaturas cuyo desarrollo está a cargo del Departamento, con el objeto de que se ajusten a los contenidos mínimos definidos en los correspondientes planes de estudio “.

Que, por lo expuesto, el Consejo Departamental de Ciencia y Tecnología debe aprobar los programas de las asignaturas a dictarse en cada ciclo lectivo.

Que por Res CD CyT N°228/15 se establecieron las pautas de presentación de los programas analíticos para las asignaturas de las carreras del Departamento de Ciencia y Tecnología.

Que por Res CS N° 171/22 se establecieron las pautas de presentación de los programas analíticos para las asignaturas de las carreras de la Universidad Nacional de Quilmes.

Que la disponibilidad de esta información es de indudable beneficio para las personas estudiantes y para el proceso tutorial de las inscripciones, y que es un requerimiento para los procesos de acreditación de las carreras ante la Comisión Nacional de Evaluación y Acreditación Universitaria (CONEAU).

Que por Resolución CD CyT N° 46/25 se aprobaron los programas analíticos de las asignaturas de las Carreras: Licenciatura en Informática.

Que es necesario aprobar el programa del Seminario:” Web Semántica y Grafos de Conocimiento.

Que la Comisión de Asuntos Académicos, Posgrado y Extensión del Departamento de Ciencia y Tecnología ha emitido despacho con criterio favorable.

Por ello,

**EL CONSEJO DEPARTAMENTAL
DEL DEPARTAMENTO DE CIENCIA y TECNOLOGÍA**

R E S U E L V E:

ARTICULO 1º: Aprobar el programa analítico de la Carrera: Licenciatura en Informática, Seminario:” Web Semántica y Grafos de Conocimiento, según se detalla en el ANEXO de la presente Resolución.

ARTICULO 2º: Regístrese, comuníquese y archívese.

Resolución CD CyT N° 239/25

PROGRAMA ANALÍTICO DE LA ASIGNATURA
Seminario: Web Semántica y Grafos de Conocimiento
Modalidad Regular

Departamento de Ciencia y Tecnología

Carrera: Licenciatura en Informática

Núcleo al que pertenece: Orientación ciclo superior

Asignaturas correlativas: No tiene

Carga horaria total: 72 hs

Carga horaria: 4 horas

Docentes: Diego Torres

Año lectivo: 2026

Objetivos

Los objetivos para quienes cursen la asignatura son:

- Adquirir ideas y técnicas para la comprensión del modelo de datos y arquitecturas básicas vinculadas a la web semántica.
- Comprender la importancia de la flexibilidad de los grafos como estructura básica para la representación de conocimiento.
- Comprender las particularidades de los lenguajes para la representación de datos estructurados como RDF, RDFS y OWL.
- Estudiar el diseño de ontologías para el modelado de dominios del conocimiento.
- Estudiar y comprender los desafíos que aparecen al construir aplicaciones más grandes, en las que el dominio a modelar es más complejo, y cuyo desarrollo lleva más tiempo.
- Conocer y construir consultas sobre grafos de conocimiento utilizando lenguajes de consulta, principalmente SPARQL.

Contenidos mínimos:

- Definición de Web Semántica
- Ontologías como recurso para la representación de conocimiento
- Fundamentos lógicos: Lógica de Descripciones
- Lenguajes para la web semántica
- Aplicaciones de la Web Social Semántica: Redes Sociales, Inteligencia Colectiva

Programa analítico

Unidad 1:

Introducción a la Web Semántica. Motivación de la aplicación de tecnologías de la Web Semántica. Datos en la Web. Introducción a la Web de Datos y a los Grafos de Conocimiento. Análisis de aplicaciones que se basan en la Web de datos (MashUps, tecnologías móviles, recomendadores, RSS).

Unidad 2:

Formas de datos en la Web 1. Datos estructurados. Ejemplos de datos publicados en la Web: documentos estructurados (XML y XSLT), microformatos (vCard), microdatos+schema.org. Reutilización de datos estructurados. Repositorios de datos abierto estructurados.

Unidad 3:

Formas de datos en la Web 2. Introducción Tecnologías de la Web Semántica. Necesidad de la semántica en la Web. Tecnologías Web Semánticas: el enfoque en capas. Metadatos: RDF – Expresión de sentencias sobre objetos. Recursos. Propiedades. Sentencias: ternas, representación basada en grafos, representación en XML. Grafos de conocimiento: definiciones y poblado.

Unidad 4:

Ontologías: RDF schema: clases, propiedades, jerarquía de clases, jerarquía de propiedades. Web Ontology Language (OWL) clases, propiedades, instancias, tipos de datos. Espacio de nombres. Clases de Clases. Clase de Equivalencia. Logica de descripciones.

Unidad 5:

La Web de Datos. URIs para nombrar cosas. Dereferenciación de URIs. Usando RDF schema. Pensar aplicaciones linkeando datos. Links relacionales, de identidad de vocabularios. El movimiento LOD (linking open data). Las 5 estrellas de LOD.

Unidad 6:

Consultas y Sparql. Frameworks para la web semántica. Búsquedas semánticas. Linking Open Data. MashUps semánticos. Ejemplos en DBPedia y Wikidata.

Unidad 7:

Consumiendo datos en la web. Criterios de selección de datos, de vocabularios. Combinación de datos. Criterios de combinación de datos.

Unidad 8.

Restricciones y limitaciones de grafos de conocimiento. Lenguajes de validación. Shapes: ShaCL y SheEx. Representación de restricciones.

Unidad 9:

Usos de IA Generativa para la Web Semántica y arquitecturas de utilización. Utilización de grafos de conocimiento para mejorar LLMs. Utilización de LLMs para la mejora de ontologías y respuestas. Configuraciones de infraestructuras para la combinación de LLMs y repositorios semánticos

BIBLIOGRAFÍA OBLIGATORIA

The Semantic Web. Scientific American, May 2001, Tim Berners-Lee, James Hendler and Ora Lassila

The Semantic Web Revisited. Nigel Shadbolt, Wendy Hall, Tim Berners-Lee (2006). IEEE Intelligent Systems.

Creating a Science of the Web. Berners-Lee, T., Hall, W., Hendler, J., Shadbolt, N. and Weitzner, D. (2006) Science, 313 (5788). pp. 769-771. ISSN 0036-8075.
<http://journal.webscience.org/2/2/creating.pdf>

Hogan, Aidan. "Web of data." The Web of Data. Springer, Cham, 2020. 15-57.

Kejriwal, Mayank, Craig A. Knoblock, and Pedro Szekely. Knowledge Graphs: Fundamentals, Techniques, and Applications. MIT Press, 2021.

The Shortest Path to the Future Web. Danny Ayers. IEEE Internet Computing Volume 10 , Issue 6 (November 2006) Pages: 76-79. ISSN:1089-7801

A Developer's Guide to the Semantic Web. Liyang Yu. Springer; 2011 edition (January 6, 2011). ISBN-10: 3642159699 ISBN-13: 978-3642159695.

Tom Heath and Christian Bizer (2011) Linked Data: Evolving the Web into a Global Data Space (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool.

The Social Semantic Web by John G. Breslin, Alexandre Passant, Stefan Decker. Springer; 1 edition (October 16, 2009). ISBN-10: 3642011713 ISBN-13: 978-3642011719.

BIBLIOGRAFÍA DE CONSULTA

A Semantic Web Primer (Cooperative Information Systems), Grigoris Antoniou, Frank van Harmelen. The MIT Press 3^o edition (2012). ISBN-13: 978-0262012102

Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Dean Allemang, (Author), James Hendler. Morgan Kaufmann (Second edition 2011). ISBN-10: 0123735564 ISBN-13: 978-0123735560

Enabling Semantic Web Services. TheWeb Service Modeling Ontology. Fensel, · Lausen, · Polleres, de Bruijn, Stollberg, · Roman, Domingue. Springer Berlin. ISBN-13 978-3-540-34519- 0 ISBN-10 3-540-34519-1

Organización de las clases:

La materia se organiza en dos tipos de clases: clases expositivas y clases de resolución de problemas. Estas últimas pueden ser de tipo trabajos prácticos o trabajos de laboratorio. Las clases teóricas serán de una carga semanal de 3 horas, en las mismas se introducirán y explicarán los temas indicados en el programa analítico. Las clases teóricas darán el marco general de discusión y se complementarán con la bibliografía obligatoria.

La mecánica de cursada para los alumnos consiste en asistir a las clases teóricas y luego resolver una serie de trabajos prácticos que tratarán los temas teóricos en diferentes formatos de actividades:

- Ejercicios teórico / práctico para que puedan profundizar los conceptos vistos en las clases teóricas y los contrapongan con la bibliografía obligatoria de la materia.
- Ejercicios prácticos para resolver aplicando los conceptos teóricos en la resolución de problemas o situaciones. Estos ejercicios pueden resolverse en parte en el horario de trabajos prácticos y en parte fuera de los horarios de la materia.
- Ejercicios de laboratorio, de mayor envergadura que los ejercicios prácticos y que requieren mayor tiempo en el análisis, discusión y resolución. Estos proyectos están diseñados para realizarse en clase durante el horario de trabajos prácticos.

Clase expositiva: Todos los temas son expuestos y explicados en clase utilizando pizarrón, presentaciones con diapositivas, videos, etc. Las clases se desarrollan en un ambiente tendiente a promover el diálogo y la formulación de

preguntas a fin de favorecer la comprensión de los diferentes contenidos disciplinares.

Clase de resolución de problemas, ejercicios y análisis de casos: El estudiantado cuenta con guías de actividades que incluyen preguntas, problemas y/o análisis de casos que se resuelven y/o discuten en el aula. En estas clases prácticas el docente atiende consultas individuales o grupales vinculadas con las actividades propuestas. Se detallan las guías prácticas utilizadas:

TP 1: La Web de Datos. Realizar el análisis para el diseño de sistema que integre diferentes fuentes de datos. Detectar desafíos y estrategias tecnológicas a utilizar. A partir de diferentes archivos vinculados a la descripción de recursos similares, definir las estrategias de unificación entre las fuentes.

TP 2: RDF y RDFs

Los objetivos de este trabajo practico son comprender la representación de conocimiento a través de la utilización de grafos. Identificar los elementos centrales del modelo RDF y RDFS. Aplicar estos conceptos al modelado de un dominio y combinarlos con vocabularios existentes (por ejemplo: FOAF, Dublin Core, SIOC).

TP 3: OWL: Comprender los elementos que definen a OWL para poder realizar el diseño de una ontología. En este diseño poder incluir diferentes tipos de propiedades y comprender las diferentes formas de aplicar reglas de inferencia y cotejo de restricciones.

TP 4: Consultas: El objetivo de este práctico es poder realizar una comprensión en la manipulación del lenguaje de consulta SPARQL. Explorar diferentes formas de realizar consultas y combinar la utilización de diferentes formas de proyección, la unión de consultas, la utilización de caminos axiomáticos. Se utilizaran diferentes endpoints particularmente el de Wikidata y la utilización de motores sparql.

TP 5: Shapes

El objetivo de este práctico es desarrollar habilidades para la construcción de restricciones sobre grafos de conocimiento. Aplicar la idea de NodeShape, restricciones de propiedades, formalización de jerarquías en las restricciones, definiciones de diferentes tipos de objetivos en el grafo de datos.

TP 6: Combinación de IA Generativa y Machine Learning

Comprender limitantes y provecho del modelo simbólico en comparativa con la utilización de otros modelos. Combinación de grafos de conocimiento con aprendizaje de máquina. Combinación de grafos de conocimiento con ia generativas. Configuración de un RAG incluyendo modelo de grafos locales. Embedding de grafos de conocimiento.

Formas de evaluación y acreditación:

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18). En la modalidad de libre a su vez, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral, un desarrollo de programa en el entorno de trabajo e instancias de evaluación similares a las realizadas en la modalidad presencial.

Cronograma tentativo

Semana	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Introducción a la Web actual y a la Web Semántica	x				
2	Representación del conocimiento. Grafos de conocimiento vs bases de datos relacionales.	x				
3	RDF y RDFS	X	x	x		
4	Practica		x	X		
5	OWL y ontologías	x	x	X		
6	Práctica		x	x		
7	Consultas sobre Grafos de Conocimiento	X	x	x		
8	Practica		X	x		
9	Restricciones y validaciones	X				
10	Practica		x	x		
11	Aplicaciones de grafos de conocimiento en la industria	x				
12	Grafos de conocimiento con Machine Learning e IA Gen	x	x			
14	Practica		x	x		
13	Parcial					x
14	Practica	x	x			
15	Recuperatorio del parcial					x
16						



Quilmes, 6 de julio de 2026

VISTO: el Estatuto de la Universidad Nacional de Quilmes, y la Resoluciones (CD) CyT N°228/15, (CS) 171/22, (CD CyT) N° 46/25 y

CONSIDERANDO:

Que de acuerdo con el artículo 43° del Estatuto de la Universidad Nacional de Quilmes “Las carreras y las diplomaturas son unidades curriculares y dependen del Departamento responsable del desarrollo de las asignaturas que constituyen el núcleo básico de los respectivos planes de estudio”.

Que el artículo 79° inc. k) del mismo Estatuto establece que los Consejos Departamentales “Aprueban y supervisan los programas de las asignaturas cuyo desarrollo está a cargo del Departamento, con el objeto de que se ajusten a los contenidos mínimos definidos en los correspondientes planes de estudio “.

Que, por lo expuesto, el Consejo Departamental de Ciencia y Tecnología debe aprobar los programas de las asignaturas a dictarse en cada ciclo lectivo.

Que por Res CD CyT N°228/15 se establecieron las pautas de presentación de los programas analíticos para las asignaturas de las carreras del Departamento de Ciencia y Tecnología.

Que por Res CS N° 171/22 se establecieron las pautas de presentación de los programas analíticos para las asignaturas de las carreras de la Universidad Nacional de Quilmes.

Que la disponibilidad de esta información es de indudable beneficio para las personas estudiantes y para el proceso tutorial de las inscripciones, y que es un requerimiento para los procesos de acreditación de las carreras ante la Comisión Nacional de Evaluación y Acreditación Universitaria (CONEAU).

Que por Resolución CD CyT N° 46/25 se aprobaron los programas analíticos de las asignaturas de las Carreras: Licenciatura en Informática.

Que es necesario aprobar el programa del “Seminario: Introducción a los LLMs y al desarrollo de aplicaciones asociadas”.



Que la Comisión de Asuntos Académicos, Posgrado y Extensión del Departamento de Ciencia y Tecnología ha emitido despacho con criterio favorable.

Por ello,

EL CONSEJO DEPARTAMENTAL DE CIENCIA y TECNOLOGÍA

RESUELVE:

ARTICULO 1º: Aprobar el programa analítico “Seminario: Introducción a los LLMs y al desarrollo de aplicaciones asociadas”, de la Carrera: Licenciatura en Informática, según se detalla en el ANEXO de la presente Resolución.

ARTICULO 2º: Regístrese, comuníquese y archívese.

PROGRAMA ANALÍTICO DE LA ASIGNATURA

Seminario: Introducción a los LLMs y al Desarrollo de Aplicaciones Asociadas

Departamento de Ciencia y Tecnología

Carrera: Tecnicatura Universitaria en Programación Informática y Licenciatura en Informática

Núcleo al que pertenece: Complementario / Orientación - Ciclo Superior

Carga horaria total: 72 hs

Carga horaria: 4 horas

Docente: Pablo Factorovich

Año lectivo: 2026

Objetivos:

Los objetivos para quienes cursen la asignatura son:



- Comprender los principios fundamentales del procesamiento de lenguaje natural y la arquitectura interna de los Modelos de Lenguaje Grandes (LLMs) desde una perspectiva intuitiva, probabilística y algorítmica.
- Diseñar e implementar arquitecturas de software eficientes que integren LLMs comerciales y de código abierto (*open-source*) para la resolución de problemas complejos de automatización y procesamiento de información.
- Dominar técnicas avanzadas de ingeniería de prompts y estructuración de datos para garantizar que las salidas de los modelos sean predecibles, seguras y acoplables a sistemas informáticos tradicionales.
- Desarrollar sistemas de Generación Aumentada por Recuperación (RAG) y agentes autónomos, integrando bases de datos vectoriales y APIs externas para mitigar las alucinaciones y la falta de actualización de los modelos.
- Evaluar de manera crítica el rendimiento, los costos, los sesgos y los riesgos de seguridad (como la inyección de prompts) en las aplicaciones desarrolladas, aplicando criterios éticos y de buenas prácticas de la industria.
- Seleccionar la estrategia técnica adecuada (RAG, *Few-shot prompting* o *Fine-tuning* conceptual) para abordar un problema de negocio o desarrollo de software según las restricciones de recursos, privacidad y hardware.

Contenidos mínimos:

Modelos estadísticos de lenguaje y tokenización de texto (BPE, WordPiece).

Representación vectorial y espacios latentes (embeddings). Arquitectura Transformer: mecanismo de auto-atención por similitud, modelos autorregresivos (Decoder-only) y parámetros de muestreo estocástico. Aprendizaje en contexto (In-Context Learning) e Ingeniería de Prompts avanzada (Chain-of-Thought, ReAct). Estructuración y validación sintáctica de salidas para sistemas tradicionales. Generación Aumentada por Recuperación (RAG): ventanas de contexto, bases de datos vectoriales e indexación por similitud. Agentes autónomos y frameworks de orquestación. Ciclo de vida de LLMs: pre-entrenamiento, ajuste fino instructivo (Instruction Tuning) y alineación. Evaluación mediante métricas



sintácticas y jueces artificiales (LLM-as-a-Judge). Seguridad y ética: inyección de prompts, jailbreaking, sesgos y privacidad. Ecosistema de modelos abiertos versus APIs comerciales.

Programa analítico:

Unidad 1: Introducción y el Concepto de Tokenización

De la Combinatoria al Texto: ¿Cómo ve el mundo un LLM? El concepto de vocabulario finito. Tokenización: Algoritmos de tokenización (BPE - Byte Pair Encoding, WordPiece). Cómo la combinatoria define la eficiencia del vocabulario.

Probabilidad de N-gramas: Modelos de lenguaje estadísticos previos a la Inteligencia Artificial profunda (Markov de primer y segundo orden).

Unidad 2: La intuición detrás de los transformers

Embeddings (Incrustaciones): Pasar de palabras discretas a un "espacio de conceptos" continuo (representación vectorial sin entrar en la geometría compleja). El Mecanismo de Atención (Atención por Similitud): Explicación intuitiva de cómo una palabra "mira" a las demás para entender el contexto (ejemplo: "banco" de sentarse vs. "banco" financiero).

Arquitectura Transformer: Encoder vs. Decoder (por qué los LLMs modernos como GPT son *Decoder-only*).

El proceso de Generación: Muestreo estocástico, Temperatura, Top-K y Top-P (conceptos puramente probabilísticos y combinatorios).

Unidad 3: Ingeniería de prompts e interacción avanzada

In-Context Learning: ¿Por qué el modelo aprende sin cambiar sus pesos?

Técnicas de Prompting: Few-shot prompting, Chain-of-Thought (Cadena de Pensamiento), ReAct (Reasoning and Acting).

Estructuración de Salidas: Forzar al LLM a responder en formatos estrictos (JSON, esquemas Pydantic) para su integración en sistemas informáticos.



Unidad 4: Desarrollo de Aplicaciones con LLMs (Orquestadores y RAG) El problema del contexto limitado y las alucinaciones.

RAG (Retrieval-Augmented Generation):

Bases de datos vectoriales (conceptos de indexación y búsqueda por similitud de cosenos intuitiva).

Inyección de contexto dinámico.

Agentes e Introducción a Frameworks: Uso de LangChain, LlamaIndex o CrewAI.

Cómo conectar un LLM a APIs externas y bases de datos SQL.

Unidad 5: Ajuste Fino (Fine-Tuning) y Evaluación Conceptual

¿Cuándo hacer Fine-Tuning vs. RAG?: Costos, beneficios y casos de uso.

Pre-training vs. Instruction Tuning: Cómo se pasa de un modelo que "completa texto" a uno que "sigue instrucciones" (RLHF conceptual).

Evaluación de LLMs: Métricas tradicionales (BLEU, ROUGE) y el paradigma moderno "LLM-as-a-Judge".

Unidad 6: Ética, Seguridad y el Futuro de los LLMs

Vulnerabilidades de Seguridad: Prompt Injection, Jailbreaking, envenenamiento de datos.

Sesgos y Toxicidad: Cómo mitigar sesgos en las respuestas del modelo.

Modelos Locales vs. APIs: El ecosistema Open Source (Llama, Mistral) frente a modelos comerciales (OpenAI, Anthropic). El factor de la privacidad de datos.

BIBLIOGRAFÍA OBLIGATORIA

- Briggs, J., & Ingham, F. (2024). Building LLM Apps: A Practical Guide to Developing Applications with Language Models. O'Reilly Media.
- Hugging Face. (2026). The Hugging Face Course: Natural Language



Processing and Local Models. Recuperado de <https://huggingface.co/learn/nlp-course>

- Jurafsky, D., & Martin, J. H. (2025). Speech and Language Processing (3rd ed. draft). Edición de distribución libre por los autores.
- LangChain & LlamaIndex Core Teams. (2026). Official Documentation and Production Architecture Guides for AI Engineers.
- Minaee, S., et al. (2025). Large Language Models: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Yao, S., et al. (2024). ReAct: Synergizing Reasoning and Acting in Language Models. ICLR.

BIBLIOGRAFÍA DE CONSULTA

- DeepLearning.AI. (2024-2026). Short Courses on Large Language Models, RAG, and AI Agents.
- Lewis, P., et al. (2020). Retrieval-Augmented Generation for

Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

- Rothman, D., & Gulli, A. (2025). Transformers and Generative AI Projects: Build Next-Generation AI Apps with Agents, RAG, and Fine-Tuning. Packt Publishing.



- Tunstall, L., von Werra, L., & Wolf, T. (2022). Natural Language Processing with Transformers. O'Reilly Media.
- Vaswani, A., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30.
- Wei, J., et al. (2023). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 / NeurIPS.

Organización de las clases:

La materia se organiza en:

Clase expositiva:

Todos los temas son expuestos y explicados en clase utilizando pizarrón, presentaciones con diapositivas, videos, etc. Las clases se desarrollan en un ambiente tendiente a promover el diálogo y la formulación de preguntas a fin de favorecer la comprensión de los diferentes contenidos disciplinares.

Clase de resolución de problemas, ejercicios y análisis de casos:

Laboratorios con pequeños entregables de código en Jupyter Notebooks que resuelvan problemas específicos de cada unidad.

Nota: ver en cada unidad la propuesta de laboratorio.

Formas de evaluación y acreditación:

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18). En la modalidad de libre a su vez, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral, un desarrollo de programa en el entorno de trabajo e instancias de evaluación similares a las realizadas en la modalidad presencial.



Cronograma tentativo

Semana	Tema/unidad	Actividad* ^[1]				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Introducción y el Concepto de Tokenización	X		X		
2	Introducción y el Concepto de Tokenización	X		X		
3	La Intuición detrás de los Transformers	X		X		
4	La Intuición detrás de los Transformers	X		X		
5	La Intuición detrás de los Transformers	X		X		
6	Ingeniería de e Prompts Interacción Avanzada	X		X		
7	Ingeniería de e Prompts Interacción Avanzada	X		X		



8	Evaluacion 1					X
9	Desarrollo de Aplicaciones con LLMs y (Orquestadores RAG)	X		X		
10	Desarrollo de con Aplicaciones LLMs	X		X		
	(Orquestadores RAG) y					
11	Desarrollo de Aplicaciones con LLMs y (Orquestadores RAG)	X		X		
12	Ajuste Fino y (Fine-Tuning) Evaluación Conceptual	X		X		
14	Ajuste Fino y (Fine-Tuning) Evaluación Conceptual	X		X		
13	Ética, Seguridad y el Futuro de los LLMs	X		X		
14	Ética, Seguridad y el	X		X		



	Futuro de los LLMs					
15	Evaluación 2					X
16	Recuperatorio					X

[1] * Indicar con X el tipo de actividad que requiere cada clase.

Hoja de firmas



Sistema: sudocu

Fecha: 13/07/2026 15:17:41

Cargado por: Silvia Beatriz Dominguez



Sistema: sudocu

Fecha: 14/07/2026 16:43:43

Autorizado por: Carla Capobianco