

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 asíncrono: Callback, promesas. Funciones asíncronas en Node: I/O archivo y red.

Práctica 6: **Asincronismo II** Formas alternativas de escribir código asíncrono 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.

Etapa 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.

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

Etapa 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.	
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 la 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