Planificacion

Programacion sobre redes

El objetivo de la cursada es lograr que los alumnos incorporen conceptos fundamentales, de que se trata y como se integra en la vida laboral. Es de extrema importancia y fundamental en todos los sistemas que vayan a desarrollar de aquí en adelante, para ello se utilizaran diferentes herramientas para poder lograr el objetivo de la materia. La misma va a girar en torno al trabajo practico integral donde van a tener que aplicar constantemente los conocimientos obtenidos en la cursada, este tiene que ser terminado y aprobado antes de que termine el ciclo electivo como condición necesaria para aprobar la materia.

El mismo va a contar con todos los temas detallados a continuación, si alguno de los contenidos detallados no esta aprobado, el trabajo practico no sera aprobado.

**Unidad 1:** **POO. Persistencia de datos. Flujos de datos.**

Esta unidad van a tener que diseñar su modelo del sistema propuesto en el tp, para luego empezar a implementarlo en el lenguaje NODE. El mismo tendra la persistencia de datos en MYSQL y tendra una API REST para poder hacer la comunicación entre las otras unidades del tp.

**Unidad 2:** **Programación distribuida.**

Al ser un sistema desarrollado en NODE el concepto de distribucion esta en su convención. Para ello contara con una API REST y un FrontEnd desarrollado en Angular**2** para poder implementar el concepto de programa distribuido. Esta unidad se evaluara con un examen al tener que desarrollar una solucion que utilice los servicios creados en el trabajo practico y utilizar los servicios desde este programa para lograr el objetivo del examen.

**Unidad 3: Concurrencia.**

Esta unidad va a ser meramente teorica, aprendiendo el concepto de concurrencia, multihilo, como funciona el sistema operativo con monoprocesadores/multiprocesadores, tabla de procesos , mutua exclusion , deadlock y algoritmo del banquero.

**Unidad 3:** **Programación concurrente.**

NODE en su concepcion es un lenguaje monohilo, pero al manejar el concepto de async de manera sin igual tiene la particularidad de “Simular” de manera sencilla y aplicable en los casos reales de la vida laboral. Para ello tendran que incorporar a su solucion las herramientas del lenguaje para poder solucionar un problema propuesto ademas de incorporar el concepto de concurrencia.

**Contenidos nodales.**

* Diagrama de Clases UML.
  + Concepto
  + Explicacion del modelo
  + Aplicacion del modelo a la solucion propuesta
  + Implementacion del modelo
* Diagrama de Secuencia UML
  + Concepto
  + Explicacion del modelo
  + Aplicacion del modelo a la solucion propuesta
  + Implementacion del modelo
* Diagrama de Entidad Relacion
  + Concepto
  + Explicacion
  + Aplicacion
  + Implementacion del modelo
* Entorno de desarrollos
  + Van a aprender las diferentes herramientas que existen en el mercado para poder elaborar una solucion que pueda ser aplicada de manera comercial.
    - Concepto de repositorio
    - Creacion de repositorio en GITHUB
    - Entendimiento de clone, pull , push, merge
    - Instalacion de paquetes iniciales
    - Investigacion de paquetes necesarios para la solucion de su desarrollo
* Programacion sobre Redes
  + Concepto
  + Concepto de API
  + Concepto de API REST
  + Elaboracion de Api REST
  + Consumir servicios REST
* Programación distribuida.
  + Explicacion de FrontEnd y Backend
  + Explicacion de Angular
  + Elaboracion de interface grafica en Angular2
  + Implementacion de Servicios REST en Angular2

**Criterios de aprobacion**

La materia contara con 3 examenes relacionados al TP y una entrega parcial del tp como segunda nota de cada trimestre.

* Primer trimestre
  + Examen: UML (clases), elaboracion diagrama para acomodar el sistema a los nuevos requerimientos
  + Tp: UML, instalaciones iniciales de la aplicación + modulos iniciales
* Segundo trimestre
  + Examen: UML (secuencia) / DER, elaboracion de diagrama para acomodar el sistema a los nuevos requerimientos, ademas de elaborar el servicio propuesto.
  + Tp: Tener la Api terminada.
* Tercer trimestre
  + Examen: Integrador. Nuevo requerimiento (hacer los 2 diagramas, mas implementacion en front y back)
  + Tp: Integracion con el FrontEnd y ultimos detalles.