

# Información del curso

Pedro O. Pérez M., PhD.

Programación avanzada  
Tecnológico de Monterrey

*pperezm@tec.mx*

06-2020

## Información del profesor

Información del profesor

## Información del curso

Intenciones educativas

Objetivos

Metodología

Evaluación

Normas de clase

Bibliografía

## Herramientas a utilizar

Herramientas a utilizar

Ambiente de trabajo

## ¿Puntos extras?

Club de Programación Competitiva

## Información del profesor

- ▶ Pedro Oscar Pérez Murueta
  - ▶ ISC Mayo 1994
  - ▶ MTI Mayo 2002
  - ▶ DCC Diciembre 2019
- ▶ Correo: [pperezm@tec.mx](mailto:pperezm@tec.mx)
- ▶ Oficina: Edificio 2, Piso 3
- ▶ Horario de asesoría: Se encuentra en la puerta de mi oficina.



## Intenciones educativas

Nuestra principal intención es que adquieras la habilidad para diseñar y programar las soluciones a esos problemas, en un lenguaje de bajo nivel (C) dependiente de la plataforma, usando correctamente el API del sistema operativo UNIX.

# Objetivos

Al finalizar este curso serás capaz de:

- ▶ Solucionar problemas de programación a nivel del sistema operativo, con énfasis en UNIX.
- ▶ Desarrollarás las habilidades necesarias para diseñar y programar soluciones con técnicas de administración de memoria, de programación concurrente para el manejo de procesos e hilos, técnicas para el manejo de funciones de entrada/salida y de mecanismos IPC para la programación con sockets.

# Metodología

Para alcanzar los objetivos que persigue la materia hemos desarrollado una estrategia organizada de la siguiente manera:

- ▶ **Autoestudio:** En algunas ocasiones deberás realizar un autoestudio previo. Los autoestudios consistirán generalmente en la lectura de un capítulo del libro de texto. Te recomendamos tomar apuntes, a mano, de los temas visto durante esta lectura.
- ▶ **Exámenes de revisión:** Al inicio de la sesión posterior a la actividad de autoestudio se aplicará un examen de revisión. Este examen durará 30 minutos máximo. **En la solución de este examen podrás utilizar los apuntes que hayas realizado durante el autoestudio, si y solo si son apuntes realizados a mano. No se aceptarán impresiones.**

- ▶ **Sesiones de clase:** El profesor desarrollará programas sencillos en donde se aplicará lo visto durante las actividades previas.
- ▶ **Ejercicios colaborativos:** Realizarás prácticas que te permitirán reforzar lo visto en clase. Toda tarea se desarrollará en equipos colaborativos usando la técnica de Pair Programming. Tendrás toda una sesión de clase para realizarla. **Si no pueden terminar la actividad durante la sesión, tendrás hasta medianoche del día siguiente para enviar la tarea, con una penalización de 50 puntos.** Toda tarea deberá estar libre de errores sintácticos (deberá compilar) para poder ser evaluada. **Cualquier tarea que no cumpla con este punto será evaluada con 10.**



- **Exámenes de tercio:** Habrá tres exámenes a lo largo del semestre (semana 5, 10 y 15). Los exámenes consistirán en desarrollar aplicaciones sencillas en un ambiente UNIX. **Si no envías la actividad durante la sesión, se considerará una calificación de NP.** Las aplicaciones deberán estar libre de errores sintácticos (deberán compilar) para poder ser evaluadas. **Cualquier examen que no cumpla con este punto será evaluada con 10.**

- ▶ **Proyecto:** Durante el semestre habrá un proyecto final en el cuál se aplicarán los conocimientos adquiridos. **Todo proyecto deberá desarrollarse individualmente.** Tu solición deberá estar libre de errores sintácticos (deberá compilar) para poder ser evaluadas. **Cualquier proyecto que no cumpla con este punto será evaluada con 10.**
- ▶ **Examen final:** El examen final cubrirá todos los temas vistos durante el semestre.

# Evaluación

Evaluación parcial		Evaluación final	
Ejercicios colaborativos	30 %	Ejercicios colaborativos	15 %
Exámenes de revisión	30 %	Exámenes de revisión	10 %
Exámenes de tercio	40 %	Exámenes de tercio	40 %
		Proyecto	15 %
		Examen final	15 %
		Foros	5 %

# Normas de clase

## Exámenes

- ▶ Los exámenes podrán ser presentados solamente en la fecha estipulada. El no presentar un examen implica una calificación de NP (No Presentó).
- ▶ El cambio de fecha de algún examen parcial deberá realizarse, a petición de los estudiantes, durante las dos primeras semanas de clase. Éste se hará sólo si se cuenta con el consenso del grupo y del profesor.

## Asistencia a clases

En lo que respecta a esta clase:

- ▶ La sesión de clase inicia 5 minutos después del horario establecido (8:35). Si no estás al inicio de la misma, se considerará que no asististe a esa sesión. Asimismo, también se considera inasistencia si te retiras, sin permiso del profesor, antes de terminar la sesión de clase.
- ▶ No podrás acreditar, bajo ningún concepto, las actividades (tareas y/o exámenes) de las sesiones a las cuales no hayas asistido. Además, será tu responsabilidad estudiar el material visto en esas sesiones.

## Tareas y Proyectos

- ▶ Toda tarea y/o proyecto tendrá su fecha y horario de entrega que es inamovible. Vencido el término de entrega no se recibirán tareas y/o proyectos.
- ▶ Todas las tareas son individuales a menos que explícitamente se pida trabajar en grupo.

## Redacción y Organización

- ▶ La mala redacción, organización y ortografía en la elaboración de tareas, proyectos, presentaciones y exámenes, será causa de penalización en la calificación correspondiente.

## Calificaciones

- ▶ Las calificaciones parciales y final se expresan en escala de uno a cien.
- ▶ La calificación mínima aprobatoria es 70 (SETENTA).

## Faltas a la Integridad Académica en Tareas, Proyectos o Exámenes

- ▶ Las faltas a la integridad académica, como la copia o tentativa de copia en cualquier tipo de examen o actividad de aprendizaje; el plagio parcial o total; facilitar alguna actividad o material para que sea copiada y/o presentada como propia; la suplantación de identidad; falsear información; alterar documentos académicos; vender o comprar exámenes o distribuirlos mediante cualquier modalidad; hurtar información o intentar sobornar a un profesor o cualquier colaborador de la institución; entre otras acciones más son consideradas faltas grave. Cuando un alumno cometa un acto contra la integridad académica, se le asignará una calificación reprobatoria a la actividad, examen, período parcial o final. La calificación reprobatoria asignada por el profesor será inapelable, y a esta sanción se sumarán las otras posibles que determine el Comité de Integridad Académica de Campus. Esto tal como lo indica el Reglamento Académico en su CAPÍTULO IX: Faltas a la integridad académica.

## Baja de materias

- ▶ Artículo 4.10: Los alumnos podrán darse de baja en una o varias materias inscritas durante el periodo académico semestral, trimestral o intensivo, **antes del último día de clases, conforme lo establezca el Calendario Escolar correspondiente.** Las materias dadas de baja no se registrarán como reprobadas. En el caso de que el alumno solicite la baja de todas sus materias, deberá pasar nuevamente por un proceso de admisión, en el que se tomará en cuenta su historial académico para la decisión. El proceso relacionado con la baja de las materias se llevará a cabo de acuerdo con las políticas administrativas definidas para ese fin.

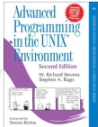
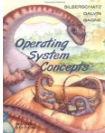



## Fechas de exámenes:

- ▶ Primer tercio: Semana 5, 10 de marzo.
- ▶ Segundo tercio: Semana 10, 21 de abril.
- ▶ Tercer tercio: Semana 15, 26 de mayo.

# Bibliografía

## Libros de texto

	<p>[STEVENS] Stevens R. W., Rago S. A. Advanced Programming in the UNIX Environment: Second Edition. Addison Wesley, 2008.</p>
	<p>[SILBERSCHATZ] Abraham Silberschatz, Greg Gagne, Peter Baer Galvin. Fundamentos de Sistemas Operativos, Cuarta Edición. Wiley, 2006.</p>
	<p>[GRIFFITHS] Griffiths, David. Head first C. O'Reilly, 2012.</p>



[https://github.com/Manchas2k4/  
advanced\\_programming](https://github.com/Manchas2k4/advanced_programming)



[https://www.remind.  
com/join/h8376f](https://www.remind.com/join/h8376f)

# Software a instalar

- ▶ Instalar Ubuntu Mate 20 LTS en una máquina virtual (VirtualBox).

## ¿Qué es el Club de Programación Competitiva?

- ▶ El Club de Programación Competitiva se trata de hacer problemas ridículamente difíciles dentro de plazos injustamente cortos. Nuestro enfoque son los concursos como ACM ICPC, VCPC, Google Codejam y Facebook Hacker Cup.
- ▶ Tenemos reuniones quincenales en las que practicamos la solución de problemas, discutimos algoritmos, lloramos por las partes que no entendimos, entendemos otras y celebramos con pizza gratis.

# ¿Qué es el Club de Programación Competitiva?

- ▶ ¿Dónde? Salón por definir.
- ▶ ¿Cuándo? Los viernes, cada quince días, a partir de 28 de febrero.
- ▶ ¿En qué horario? 2 a 4 pm.

## ¿Cuál es la oferta?

- ▶ Por cada problema B o C que resuelvas en el Club de Programación Competitiva, tendrás un punto extra en tu calificación final... con las siguientes condiciones:
  - ▶ Deberás explicar el problema resuelto, a más tardar, antes de que empiece el siguiente concurso.
  - ▶ **Los puntos extras se aplicaran, SI Y SÓLO SI APRUEBAS LA MATERIA.**

# UNIX

- ▶ UNIX es un sistema operativo de tiempo compartido que nos brinda una serie de servicios:
  - ▶ Controla los recursos del hardware.
  - ▶ Controla los dispositivos periféricos (discos, terminales, impresoras, etc.)
  - ▶ Permite a distintos usuarios compartir recursos y ejecutar sus programas.
  - ▶ Proporciona un sistema de archivos que administra el almacenamiento de información (programas, datos, documentos, etc.)





- ▶ El sistema de archivos de UNIX se caracteriza por:
  - ▶ Poseer una estructura jerárquica.
  - ▶ Realizar un tratamiento consistente de los datos de los archivos.
  - ▶ Poder crear y borrar archivos.
  - ▶ Permitir un crecimiento dinámico de los mismos.
  - ▶ Tratar los dispositivos y periféricos como si fueran archivos.
  - ▶ Se pueden manejar varios discos físicos con diferentes sistemas de archivos (tenemos muchos a nuestra disposición: ext, vfs, nfs, ufs, jfs).
  - ▶ Un sistema de archivos se compone de una secuencia de bloques lógicos, cada uno de los cuales tiene un tamaño fijo (que varía de un sistema de archivos a otros).

- ▶ La estructura de un sistema de archivos en UNIX/Linux se puede definir:
  - ▶ **Sector de arranque:** Ocupa la parte del principio del sistema de archivos, típicamente el primer sector, y puede contener el código de arranque que se encarga de buscar el sistema operativo y cargarlo en memoria para inicializarlo.
  - ▶ **Superbloque:** Describe el estado de un sistema de archivos.
  - ▶ **Nodos i:** Contiene la información necesaria para que un proceso pueda acceder al archivo.
  - ▶ Bloques de datos.