

Programa de Entrenamiento Académico (PEA) LAECI 2016

El PEA en Ciencias Informáticas de LAECI, se basa en una propuesta académica formulada especialmente para personas alto CI. El mismo, se encuentra dividido en áreas de conocimiento, las cuáles, son de cursado y asistencia obligatoria.

OBSERVACIONES: Los estudiantes podrán cursar un máximo de 3 (tres) asignaturas de forma simultánea y un mínimo de 1 (una) materia por mes, para mantener la condición de alumno regular. La duración total del Programa de Entrenamiento Académico (PEA) en LAECI, está supeditada a los tiempos de respuesta del alumno y la disponibilidad de las asignaturas. No obstante, se estima que el total del programa podría completarse en un lapso no inferior a 11 (once) meses y, en un escenario ideal, no mayor a 16 (dieciséis) meses.

COD.	Área de conocimiento	Correl.	Duración	Lab.
1	Introducción a la Ingeniería de Software	s/c	3 meses	SI
2	Inmersión en el paradigma de la programación orientada a objetos	1	2 meses	SI
3	Razonamiento lógico y métodos científicos	1 - 2	1 mes	SI
0.1	Estandarización del código fuente	s/c	2 semanas	-
4	Técnicas avanzadas de programación (<i>Refactoring</i>)	1 - 2 - 0.1	1 mes	SI
5	Ingeniería de núcleo (Ingeniería de Software avanzada)	1 - 2 - 4 - 0.1	4 meses	SI
6	Introducción al <i>Shell Scripting</i>	s/c	3 meses	SI
7	Introducción a la administración de Sistemas GNU/Linux	s/c	2 meses	SI
8	Gestión de Proyectos y Análisis de Requerimientos	s/c	2 meses	-
9	Seguridad Informática en la Ingeniería de Software	1	2 meses	SI
10	Ingeniería de bajo nivel	1 - 2 - 3 - 4 - 5 - 0.1	1 meses	SI

Contenidos

COD. 1 - Introducción a la Ingeniería de Software

1. Introducción a la arquitectura de Software. Diferencia entre estilo arquitectónico, patrón arquitectónico y patrón de diseño
2. Estructura de archivos y directorios en aplicaciones Web modulares bajo el patrón arquitectónico MVC
3. Montaje y configuración de Virtual Hosts en Apache. Laboratorio.
4. URL virtuales. Configuración del archivo .htaccess de Apache. Laboratorio.
5. Interfaces Gráficas de Usuario (GUI) en HTML y CSS. Sustituciones estáticas directas. Laboratorio.
6. Introducción a bases de datos y el lenguaje SQL de MySQL/MariaDB. Laboratorio.
7. Sistemas ABM/CRUD en MVC. Introducción al concepto de «objetos en estado puro». Generación de Modelos, métodos y propiedades del objeto. Generación de vistas auxiliares y recursos básicos del controlador. Laboratorio.
8. Capas de abstracción a bases de datos. PDO. Interfaces de consultas a datos
9. El recurso agregar. Persistencia de objetos. Laboratorio.
10. El recurso ver. Recuperación de objetos. Laboratorio.
11. El recurso editar. Modificación de objetos. Laboratorio.
12. El recurso eliminar. Destrucción de objetos. Laboratorio.
13. Colecciones de objetos
14. Sustituciones iterativas. Laboratorio.
15. Seguridad informática: seguridad por diseño, saneamiento de código, saneamiento de datos vs. validación. Laboratorio.
16. Sesiones de usuario y restricción de acceso. Introducción a la criptografía básica. Laboratorio.

COD. 2 – Inmersión en el paradigma de la programación orientada a objetos

1. Propiedades simples vs propiedades compuestas. Objetos compuestos vs objetos compositores. Patrón de diseño compuesto. Los métodos save() y get() en objetos compuestos. Mapeo relacional de objetos compuestos. Laboratorio.
2. Propiedades colectoras. Patrón de diseño de agregación. Objetos compositores dependientes. El método get() auxiliar en compositores dependientes. Persistencia y recuperación de relaciones “muchos-a-uno”. Mapeo relaciones de objetos compositores de dependencia. Laboratorio.
3. Conector Lógico Relacional: persistencia y recuperación de relaciones “muchos-a-muchos”. Mapeo relacional de conectores lógicos. Laboratorio.
4. Clonación de objetos y multiplicación. Propiedades al vuelo en la multiplicación de objetos. Laboratorio.

COD. 3 – Razonamiento lógico y métodos científicos

1. Propedéutica. Lógica. Lógica argumentativa. Racionalismo, empirismo y pragmatismo. Premisas falsas y falacias. Paradojas y contradicciones. Axiomas. Silogismo clásico.
2. Valores de verdad y lógica proposicional.
3. Metodología científica para la elaboración de teorías: el MC-14.
4. Mecanismos de observación, análisis y deducción lógica.
5. Elaboración de informes científicos. Laboratorio.

COD. 0,1 – Estandarización del código fuente (seminario)

Normalización de nombres. Espaciado y tabulación. Documentación y auto-documentación de código. Taxonomía del proceso de desarrollo. Responsabilidad de las funciones. *Settings vs config*. Laboratorio.

COD. 4 – Técnicas avanzadas de programación: Refactoring

1. Introducción al *refactoring*. Principios y reglas.
2. Variables temporales mal implementadas: acciones concretas, reasignación de parámetros, reutilización. Laboratorio.
3. Sobre-saturación de parámetros. Laboratorio.
4. Expresiones de longitud ininteligible. Laboratorio.
5. Extracción de código en la multiplicidad de responsabilidades. Laboratorio.
6. Duplicidad de código: redundancia de clase. Laboratorio.
7. Duplicidad de código: redundancia de múltiples clase sin herencia. Laboratorio.
8. Duplicidad de código: redundancia de múltiples clase con herencia. Laboratorio.

COD. 5 – Ingeniería de núcleo (Ingeniería de Software avanzada)

1. Entiendo el *core* de una aplicación. Introducción a la ingeniería de Software basada en núcleos. Núcleo -o kernel- vs Framework. Archivos básicos del *core*. Laboratorio de Objetos genéricos: Standard Object. Herencia y clases abstractas.

2. Laboratorio de colecciones de objetos genéricas: patrón de diseño *Singleton*
3. Laboratorio de fabricación de objetos con *Factory*.
4. Laboratorio de creación de servidores de archivos estáticos.
5. Laboratorio de generación de APIs REST genéricas.
6. Laboratorio de creación de un conector lógico universal.
7. Librería de métodos auxiliares para la generación de Interfaces Gráficas.

COD. 6 – Introducción al *Shell Scripting*

1. Introducción al scripting. *Hashbang* y *Shebang*. Lenguajes de *scripting* en sistemas GNU/Linux. *Scripts* ejecutables. Generación de nuevos comandos del sistema operativo. Laboratorio.
2. *Scripting* en GNU Bash. Laboratorio.
3. *Scripting* en Python. Laboratorio.
4. *Scripting* en PHP. Laboratorio.
5. Creación de páginas MAN. Laboratorio.
6. Creación de demonios del sistema. Laboratorio.
7. Generación de TUIs (*Text User Interfaces*). Laboratorio.
8. Empaquetado, distribución e instalación con *dpkg*. Laboratorio.
9. Introducción al lenguaje C. Compilación de código fuente. Laboratorio.

COD. 7 – Introducción a la administración de Sistemas GNU/Linux

1. Sobre la actualización del sistema y los índices de repositorios.
2. Administración de usuarios y grupos. Laboratorio.
3. Sistema de archivos.
4. Protocolo de conexión SSH. Laboratorio.
5. El archivo `/etc/hosts`

6. Obteniendo información del sistema físico y virtual. Laboratorio.
7. Cron, crontab y tareas programadas.
8. Iptables, nmap y herramientas de red. Laboratorio.
9. Demonios del sistema. Laboratorio.
10. Instalación y configuración de plataformas GLAMP. Laboratorio.
11. Seguridad del sistema operativo y el servidor. Laboratorio.
12. Administración de Apache. Laboratorio de adquisición, instalación y configuración de certificados TLS (antes: SSL).
13. Sistemas de control de versiones. Laboratorio.
14. Análisis de logs. Laboratorio.
15. Enjaular usuarios con chroot. Laboratorio.

COD. 8 – Gestión de Proyectos y Análisis de Requerimientos

1. Introducción a las metodologías ágiles y predictivas. Organización del proyecto con *Scrum*. Roles, artefactos y ceremonias.
2. *Backlog* de producto y definición de historias de usuario.
3. Relevamiento de requerimientos y criterios de aceptación (observación del “qué”).
4. Análisis de requerimientos y de los criterios de aceptación (análisis del “qué”).
5. Definición y asignación de tareas (deducción del “cómo”).
6. Taller de estimación de esfuerzos.
7. Taller de análisis.

COD. 9 – Seguridad informática en la Ingeniería de Software

1. Inteligencia artificial en el saneamiento de datos. Laboratorio de creación de un sistema de codificación de caracteres alfanuméricos.

2. Emulación de tokens de seguridad en el registro temporal de usuarios. Laboratorio.
3. Sistemas de autenticación seguros. Discusión de las propuestas del «Open Web Application Security Project» (OWASP). Laboratorio.
4. Sistemas de recuperación de contraseñas. Laboratorio.
5. Cross Site Scripting y editores WYSIWYG. Laboratorio de creación de editores WYSIWYG alternativos.
6. Carga de archivos al servidor desde el navegador Web. Laboratorio.

COD. 10 – Ingeniería de bajo nivel

Práctica de laboratorio final obligatoria, para la graduación del Programa de Entrenamiento Académico (PEA) de LAECI.