



Instituto de Educación Superior N° 7

Carrera: Técnico Superior en Desarrollo de Software. Res. Ministerial N° 2120/16

Año lectivo: 2022

Unidad Curricular: Ingeniería de Software II

Formato: Laboratorio

Régimen de cursado: ANUAL

Curso: 2 Año

Profesora: SUSANA ISABEL ARCE

FUNDAMENTACIÓN

Esta unidad curricular tiene la finalidad de introducir al estudiante en el trabajo de Ingeniería, llevando a cabo proyectos con la utilización de métodos y la ayuda de herramientas propias de su ámbito de acción. Reconocer, diferenciar y validar las necesidades que justifican sus proyectos. Documentar escenarios y propuestas que respondan a dichas necesidades contemplando las demandas de artefactos requeridos por las empresas de la región.

Los estudiantes deberán comprender que las normas ISO son el marco de referencia que contiene los procesos, actividades y tareas involucradas en el desarrollo, operación, soporte y mantenimiento de un producto software, para proporcionar una estructura común que responda a los estándares internacionales.

Se introducirá a los estudiantes en la implementación de un simulacro a menor escala de un producto utilizando el entorno de desarrollo Arduino, trabajando conjuntamente con alumnos de 3er. año de la tecnicatura de Energías Renovables que ya adquirieron los conocimientos de este entorno de programación al cursar en 2do. año Inmótica. Con esta experiencia participarán en la elaboración de un Proyecto y su ejecución bajo una propuesta de trabajo en equipo articulando con otra carrera.

PROPÓSITOS

El desarrollo de esta unidad curricular tiene como finalidad afianzar en los estudiantes la capacidad de diseñar artefactos de software que resuelvan problemas planteados.

Analizar críticamente la eficiencia y mantenibilidad de diseños alternativos. Diseñar las estructuras de datos e interfaces que las mismas utilicen. Analizar y discutir su eficiencia y escalabilidad. A partir del diseño, analizar clases de equivalencia y diseñar esquemas de prueba. Documentar el diseño.

Articular la tecnicatura de Software con la de Energías Renovables, intercambiando y enriqueciendo los saberes de los estudiantes, interdisciplinando a través de una experiencia sustentada por conocimientos que poseen en común.

Para concretar estos propósitos, las clases y las actividades planteadas en forma presencial también se compartirán de forma virtual en la plataforma educativa del IES N° 7 y nos continuaremos comunicando por chat o WhatsApp.

OBJETIVOS

- Interpretar críticamente las especificaciones recibidas.
- Interpretar la arquitectura del sistema en el cual se inserta la asignación.
- Aplicar patrones de diseño a la problemática abordada si corresponde.
- Diseñar la solución de los problemas a resolver.
- Representar el diseño.
- Verificar el diseño.
- Intercambiar saberes al formar un equipo de trabajo con estudiantes de otra carrera.
- Informar sus posibilidades de acceso a los recursos digitales disponibles, así como los inconvenientes que puedan presentarse en toda su trayectoria educativa.

CONTENIDOS

Principios generales de diseño: descomposición, desacoplamiento, cohesión, reusabilidad, portabilidad, testeabilidad, flexibilidad, escalabilidad. Patrones de diseño. Arquitecturas de software: concepto de vistas, arquitecturas distribuidas, “pipe-and-filter”, “model-view-controller”.

Diseño orientado a objetos. Diseño estructurado. Diseño orientado al reuso de componentes, incorporación de elementos disponibles al diseño. Diseño de interfaces con el usuario. Prototipos rápidos.

Diseño orientado a objetos, encapsulamiento y ocultamiento de información, separación entre comportamiento e implementación, clases y subclasses, herencia (sustitución), polimorfismo (subtipos vs. herencia), jerarquías de clases, clases colección y protocolos de iteración.

Distinción entre validación y verificación. Enfoques estáticos y dinámicos.

Fundamentos de “testing”, testeo de caja negra y de caja blanca. Pruebas funcionales: generación de casos o datos de prueba, clases de equivalencia. Pruebas estructurales: pruebas estáticas, pruebas dinámicas, cobertura de la prueba. Otro tipo de objetivos: verificación de usabilidad, confiabilidad, seguridad. Registro de fallas, seguimiento de fallas e informes técnicos.

Prueba unitaria, de integración, validación y prueba del sistema. Desarrollo conducido por el testeo. Refactorización del código. Testeo de regresión. Verificación y validación de artefactos que no constituyen código: documentación, archivos de ayuda, material de capacitación. Inspecciones, revisiones cruzadas, auditorías.

Herramientas para desarrollo en ambientes web. “Frameworks” de aplicaciones y su utilización.

Práctica Formativa

Como parte de la forma de adquirir estos aprendizajes y demostración práctica de los resultados alcanzados, los estudiantes tienen que realizar en un mínimo del 33%, las siguientes actividades:

Diseñar artefactos de software (clases, objetos, métodos, algoritmos, tablas) que resuelvan problemas planteados. Analizar críticamente la eficiencia y mantenibilidad de diseños alternativos.

Relacionar situaciones con patrones de diseño. Analizar diversos tipos de arquitectura de sistemas de software, discutiendo sus propiedades de calidad (escalabilidad, portabilidad, seguridad, mantenibilidad). Construir prototipos rápidos con herramientas

sencillas.

Planificar y diseñar casos y conjuntos de datos para prueba de artefactos dados, respondiendo a objetivos y requisitos de cobertura. Implementar pruebas de programas y pequeños sistemas utilizando herramientas y creando ambientes necesarios, realizar los procesos y revisar los resultados para generar informes de fallas.

Desarrollar proyectos grupales durante los cuales se simulen condiciones similares a las del trabajo profesional y en los que cada uno aporte componentes que deben integrarse en el producto final.

METODOLOGÍAS DE TRABAJO

Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.

Facilitar el contacto directo con materiales e instrumentos, al llevar a cabo actividades prácticas, para contribuir a la formación de las competencias para el trabajo experimental como: identificación, manejo y control de variables y datos relevantes, planteamiento de hipótesis, trabajo en equipo.

Fomentar actividades grupales que propicien la comunicación, el intercambio, argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes al interior de la carrera y entre carreras.

Observar y analizar fenómenos y problemáticas propias del campo ocupacional.

ACTIVIDADES

Producir diagramas de clase a partir de problemas correspondientes a diversos dominios. Analizar y discutir sus propiedades y corrección. Representar situaciones determinadas utilizando diagramas UML u otras técnicas. Analizar y discutir sus características y defectos. Modelizar y especificar casos de usos a partir de descripciones de situaciones realistas. Documentar escenarios. Revisar documentos de requerimientos de software utilizando buenas prácticas para determinar su calidad. Realizar revisiones cruzadas de especificaciones. Utilizar las herramientas de soporte en todas las actividades que realicen.

Participar en la ingeniería de una aplicación elaborada por los alumnos de 3er. año de Prácticas profesionalizantes II “Aplicación para celular”, efectuando el registro para insumo de estudio de casos para los futuros estudiantes y con la finalidad de continuar el mismo proyecto al siguiente año en sus prácticas, superando el proyecto inicial.

Trabajo colaborativo interactuando con estudiantes de otra carrera.

EVALUACIÓN

Prácticas de laboratorio.

Entrega de reporte de Tópicos investigados en el formato acordado entre estudiante y profesor.

Exposición de temas seleccionado entre estudiantes y el profesor.

Defensa de proyectos.

Condición presencial con el 75% de asistencia.

Regularizar el espacio con un mínimo de 6 (seis) puntos o más.

Acceso a 2 instancias de exámenes finales consecutivos al año de cursada.

Promoción directa: presentación y aprobación del 100% de los trabajos. Aprobación de las defensas y exposiciones. Coloquio con la defensa del Proyecto. Aprobar todas las instancias evaluativas con 8 (ocho) o más puntos.

BIBLIOGRAFÍA

BORRERO, L. Tecnologías de la Información En Internet. Editorial Norma Colombia. 2003

PRESSMAN, R. S. Ingeniería del Software – Un enfoque práctico. 7ma. Edición McGraw Hill. Interamericana Editores. 2010

WEITZENFELD, A. Ingeniería de software orientada a objetos. Cengage learning editores. México. 2005.

<https://www.arduino.cc/>

PROF. PROG. SUSANA ISABEL ARCE
