

Asignatura: **Ingeniería de Software y Hardware**

Código: 10-09805

RTF

8

Semestre: 5

Carga Horaria

96

Bloque: TA

Horas de Práctica

48

Departamento: Computación

Correlativas:

- Programación Avanzada
- Electrónica Digital 2

Contenido Sintético:

- Historia y visión general. Herramientas relevantes, estándares y/o restricciones de ingeniería.
- Gestión de proyectos.
- Riesgo, confiabilidad, seguridad y tolerancia a fallos.
- Procesos de hardware y software. Análisis y elicitación de requisitos.
- Especificaciones del sistema.
- Diseño y evaluación arquitectónica del sistema.
- Diseño concurrente de hardware y software.
- Integración, pruebas y validación de sistemas.
- Mantenimiento, sostenibilidad, manufacturabilidad

Competencias Genéricas:

- CG2: Concebir, diseñar y desarrollar proyectos de ingeniería (sistemas, componentes, productos o procesos).
- CG3: Gestionar -planificar, ejecutar y controlar- proyectos de ingeniería (sistemas, componentes, productos o procesos).
- CG4: Utilizar de manera efectiva las técnicas y herramientas de aplicación en ingeniería.
- CG7: Comunicarse con efectividad.
- CG8: Actuar con ética, responsabilidad profesional y compromiso social, considerando el impacto económico, social y ambiental de su actividad en el contexto local y global.
- CG10: Actuar con espíritu emprendedor.

Aprobado por HCD: 1042-HCD-2023

RES: Fecha: 27/11/2023

Competencias Específicas:

- CE1.2 Analizar, especificar, diseñar y proyectar arquitectura de sistemas informáticos.
- CE1.4 Implementar y mantener sistemas informáticos.
- CE4.3: Analizar, diseñar, programar, implementar, probar, depurar y evaluar hardware y software para sistemas de computación de propósitos específicos.
- CE4.4: Analizar, diseñar, programar, implementar, probar, depurar y evaluar hardware y software para sistemas de computación de propósitos generales.
- CE4.11 Analizar, proyectar y desarrollar proyectos de software y sistemas conjuntos de hardware y software haciendo uso de conceptos, métodos y herramientas de gestión de proyectos, ingeniería de software, elicitación, análisis, especificación y validación de requerimiento.
- CE8 Certificar el funcionamiento, condición de uso o estados de Sistemas de Procesamiento de Señales, Sistemas Embebidos, Sistemas Computarizados de automatización y control, Sistemas Conjuntos de Hardware y Software.

Presentación

La asignatura Ingeniería de Software y Hardware pertenece al 5(quinto) semestre (3 año) de la carrera de Ingeniería en Computación.

Al inicio de éste espacio curricular el estudiante ya cuenta con bases sólidas de algoritmos, programación y de estructuras de datos que sirven de base para el desarrollo de la presente propuesta educativa. En ella se busca conocer, descubrir, aprender, implementar, experimentar, emplear y mantener procesos y metodologías de desarrollo, como así también buenas prácticas de ingeniería ampliamente usados y probados en la industria del desarrollo de software y hardware.

Durante el transcurso de la asignatura exploramos las dificultades de producción de software y hardware a gran escala, su configuración, su administración, su control y mantenimiento; para finalmente arribar en implementaciones de procesos de ingeniería que brinden soluciones a dichas problemáticas.

La asignatura está diseñada desde un enfoque constructivista donde los estudiantes deben organizarse alrededor de metodologías ágiles para ir cumpliendo los retos y desafíos que proponemos entregar valor al final de cada sprint de 2 semanas de duración.

La cátedra ha establecido pautas para conformar un esquema de autonomía alineada, donde los docentes establecen los lineamientos y los equipos de trabajo tienen autonomía para resolver problemáticas reales y desafíos de la industria en términos de Software y Hardware. En términos más concretos, los docentes explican, ejemplifican y facilitan herramientas, patrones, métricas, etc, pero son finalmente los alumnos quienes deben implementarlas, adaptarlas, configurarlas, probarlas y mantenerlas de acuerdo a cada singular problemática que se desea resolver.

Contenidos

Módulo I. Ingeniería de Software y Hardware

Objetivo: manejar el contexto de la Ingeniería de Software y Hardware, el lenguaje y la terminología asociada a la disciplina.

Unidad 1. Introducción

Contenido: Introducción a la Ingeniería de Software.

Unidad 2 . Procesos de Software

Contenido: Modelos de Desarrollo de Software y Hardware Tradicionales y Metodologías Ágiles. Sus comparaciones. Ventajas y Desventajas. Actividades del Proceso: Especificación del proceso, Diseño e implementación del software, Validación y Verificación del Software y Hardware, evolución del software. Herramientas.

Unidad 3: Metodologías Ágiles

Contenido: Modelos Ágiles de Desarrollo. Manifiesto Agile. Modelos Modernos de Metodologías Ágiles. Combinaciones de Metodologías de acuerdo al contexto y naturaleza de los Proyectos de Ingeniería de Software y Hardware. Métricas. Herramientas. Indicadores. Artefactos. Acuerdos y Definiciones. Buenas Prácticas. Retos y Desafíos Modernos.

Módulo II. Requerimientos y Administración de la Configuración

Objetivo: manejar el contexto de la Ingeniería de Requerimientos, el lenguaje y la terminología asociada a esta disciplina. Establecer las pautas y lineamientos de la administración de la configuración.

Unidad 4: Requerimientos del Software y Hardware

Contenido: Requerimientos funcionales, no funcionales y de dominio. Requerimientos de Usuario. Requerimientos de Sistema. Especificaciones de la Interfaz. Procesos de la Ingeniería de Requerimientos. Adquisición, Elicitación y Análisis de requerimientos. Validación de requerimientos. Gestión de requerimientos.

Unidad 5: Administración de la Configuración

Objetivo: Identificar y establecer los esquemas necesarios para una correcta gestión de las configuraciones, sus implicancias, los procesos y las herramientas de Control de Versiones. Contenido. Gestión del Versionado. Sistemas de Buildings. Manejo de los Cambios

Módulo III. Diseño, Modelado e Implementación.

Objetivo: desarrollar habilidades y manejar las técnicas de diseño, modelado y construcción de sistemas de software y hardware

Unidad 6: Modelos del Sistema Software y Hardware

Contenido: Modelos de Contexto. Modelos de Comportamiento. Modelos de datos. Modelos de Objetos. Métodos Estructurados. Modelado de Sistemas. UML.

Unidad 7: Diseño arquitectónico del Software

Contenido: Decisiones del diseño arquitectónico. Organización del sistema. Estilos de descomposición modular. Patrones de Arquitecturas Tradicionales. Patrones Modernos. Arquitectura distribuidas.

Unidad 8: Sistemas Basados en Componentes

Contenido: Component-Based Software Engineering (CBSE). Modelos Basados en la reutilización. Procesos y Flujos de Trabajo con Componentes. Patrones de Diseño. Patrones Tradicionales de Diseño. Patrones Creacionales, Estructurales y Conductuales. Patrones Modernos de Diseño.

Módulo IV. Pruebas y Distribución de Software y Hardware

Unidad 9: Pruebas del Software

Contenido: Pruebas Unitarias. Pruebas de Integración. Pruebas de Sistema. Pruebas de Aceptación. Pruebas de UI. Diseño de Casos de Prueba. Escenarios. Automatización. Ambientes de pruebas. Trackeo y seguimiento de Errores, Fallas y Defectos.

Unidad 10: Distribución y Puesta Productiva.

Contenido: Gestión de las Versiones Productivas. Canales Alfa, Beta. Canary Releases. Gestión Remota de las funcionalidades. Dark Launch. Técnicas de Distribución para grandes desarrollos y múltiples equipos. Release Train. Técnicas de Distribución para productos de software/hardware con millones de usuarios.

Metodología de enseñanza

Las clases incluyen actividades donde priman las teóricas, prácticas y de laboratorio alternativamente. Las actividades teóricas se realizan a través de exposición dialogada, orientadas a aproximar a los estudiantes a las metodologías de desarrollo de software y hardware y su ámbito de aplicación.

Durante el desarrollo de los Trabajos Prácticos se realizan actividades que le permiten al estudiante poner en práctica las habilidades y verificar los criterios y técnicas de modelado, diseño y pruebas. Tanto el dominio del problema, como la naturaleza de las soluciones, el alcance y el enfoque del proyecto son ideados y proporcionados por los estudiantes, consiguiendo mayor compromiso y un sin fin de casuísticas y variedad de proyectos únicos para la cátedra. Asimismo esto permite trabajar con problemáticas reales, con casos de usos concretos y sacados de la realidad de los estudiantes y su entorno, pudiéndose repetir dichos escenarios en su vida profesional.

Respecto de los Prácticos de Laboratorio y el Proyecto Final de la Asignatura, se ha diseñado e implementado un sistema de seguimiento al estilo de la metodología ágil, donde dividimos las clases, los prácticos y las entregas de los trabajos en 7 sprints /7 entregas; es decir una entrega cada 2 clases. Ésto nos permite un seguimiento mucho más profundo y un trackeo más individual, con el objeto de identificar posibles desviaciones al plan original donde se intenta y se logra establecer un objetivo tanto individual como en equipo por sprint, de manera que sea bien claro y acotado. Con ésta metodología original, hemos innovado en la participación y compromiso individual y colectivo desde el sprint 1, generando más motivación y entusiasmo para lograr terminar la propuesta educativa y el programa de la materia. Tanto la cantidad de sprints como el alcance puede ir variando de cohorte en cohorte con el objetivo de ajustar mejor la propuesta a los cambios generacionales de las metodologías.

Evaluación

La evaluación de los conocimientos disciplinares se realizará a través de dos parciales a lo largo del semestre y las evaluaciones de las habilidades y competencias se realizarán de forma continua al finalizar cada sprint/bloque con las entregas y exposiciones de los trabajos realizados que vayan aportando valor a su trabajo final.

Al finalizar, los equipos de desarrollo deberán exponer una presentación final de su proyecto desde el inicio hasta su estado actual, luego de esas iteraciones y haciendo una

distribución/release del software/hardware funcionando con las características/features comprometidas en las etapas tempranas del proyecto.

Las actividades propuestas están diseñadas teniendo en cuenta los resultados de aprendizaje, para que su conclusión sea indicador de haber alcanzado un nivel de desarrollo aceptable de las competencias propuestas.

La calificación final será una combinación de los exámenes, las presentaciones al final de los sprints y la presentación final.

Condiciones de aprobación

Calificación:

La calificación se obtendrá a través del siguiente polinomio:

$$\text{CALIFICACIÓN} = 0,2xP1 + 0,2P2 + 0,4xS + 0,2PF$$

Donde:

P1: Es la nota del primer examen parcial.

P2: Es la nota del segundo examen parcial.

S: Es el promedio de las calificaciones de las actividades prácticas entregadas al final de los Sprints donde se evalúa a nivel individual y grupal pudiendo diferir entre integrantes del mismo equipo por cuestiones relacionadas con la exposición, el compromiso, la participación, la colaboración con otros equipos, el mentoreo, el liderazgo, etc.

PF: Es la calificación de la presentación final donde se evalúa no solo la exposición, sino también la capacidad de resolución de problemas y los casos de éxito y fracaso; valorando positivamente la capacidad de resiliencia y tolerancia a dichas condiciones adversas. Aquí se busca motivar, premiar y desarrollar esta habilidad muy necesaria para formar futuros emprendedores y líderes

Requisitos para alcanzar la regularidad.

- 80% de asistencia.
- Rendir y aprobar los dos parciales con un 60% o más, el alumno podrá recuperar sólo un parcial.
- Aprobación del 100% de las actividades prácticas propuestas al final del Sprint.
- Aprobación de la Presentación Final del Trabajo de Laboratorio.

Requisitos para alcanzar la Promoción.

- Cumplir los requisitos de regularidad
- Tener aprobadas las materias correlativas
- La calificación final debe ser igual o mayor que 7.

Los alumnos que no alcancen la condición de regular al finalizar la Presentación Final serán considerados libres.

Actividades prácticas y de laboratorio

Se espera que los alumnos desarrollen dos trabajos prácticos y un trabajo final durante el transcurso de la asignatura. Para ello se emplea una metodología de revisión semanal y

entrega parcial cada dos semanas y el desarrollo de los trabajos será en grupos de 4 estudiantes como máximo, pudiendo modificarse si la cátedra lo considera necesario. Esto tiene por objetivo que los alumnos empleen técnicas de desarrollo ágil que son prácticas estándares en la industria.

El objeto sobre el cual se desarrollaran las actividades serán de libre elección de los estudiantes, pudiendo tratarse de aplicaciones para teléfonos, tablets o de escritorio, sitios web, mejoras sobre software desarrollado en alguna otra materia e incluso sobre software libre de cualquier índole. En todos los casos, la propuesta será analizada por el docente para determinar que cumpla con lo esperado para la materia según complejidad, posible duración del proyecto, alcance, entre otros parámetros y puntos básicos comunes.

Habiendo definido el tema del trabajo, cada grupo deberá presentar tres documentos.

- **Plan de Gestión de las Configuraciones.**
- **Documento de Requerimientos.**
- **Casos de Pruebas de Sistema**

En las secciones siguientes se detallarán los requisitos mínimos para cada uno de los documentos.

Plan de Gestión de las Configuraciones

Siguiendo los planes de administración y control de las configuraciones mostrados en clase, cada grupo deberá elaborar y presentar un **Plan de Gestión de las Configuraciones** que incluya como **mínimo** los siguientes puntos:

1. Dirección y forma de acceso a la herramienta de control de versiones.
2. Dirección y forma de acceso a la herramienta de integración continua.
3. Dirección y forma de acceso a la herramienta de gestión de defectos.
4. Esquema de directorios y propósito de cada uno.
5. Normas de etiquetado y de nombramiento de los archivos.
6. Plan del esquema de ramas a usar.
7. Políticas de fusión de archivos y de etiquetado de acuerdo al progreso de calidad en los entregables.
8. Forma de entrega de los “releases”, instrucciones mínimas de instalación y formato de entrega.
9. Change Control Board. Se debe incluir el propósito, la lista y forma de los integrantes del equipo y su rol en la CCB, la periodicidad de las reuniones, etcétera.
10. Herramienta de seguimiento de defectos usada para reportar los defectos descubiertos y su estado. Forma de acceso y dirección.
11. Cualquier otra información relevante.

Documento de Requerimientos

Siguiendo los modelos expuestos en clase, cada grupo debe elaborar un **documento de requerimientos**.

- El documento debe presentar el detalle de los requerimientos funcionales y no funcionales.
- También, deberá incluir diagramas de casos de uso, diagramas de actividades, diagramas de secuencia y cualquier otro diagrama que considere necesario para mejorar el detalle y la explicación de los requerimientos del software a construir.

- Alternativamente, se debe generar una matriz de trazabilidad entre los requerimientos y los casos de uso generados.

Casos de Pruebas de Sistema

Además, se deben generar los **Casos de Prueba de Sistema** contra los requerimientos funcionales y no funcionales. Además de los casos de prueba de uso normal, incluir casos de prueba alternativos que prueban valores límites o inusuales y que tratan de generar errores no esperados.

Diseño, Implementación y Pruebas

En éste trabajo práctico, cada grupo presentará una serie de documentos detallando el diseño e implementación de su proyecto además de las pruebas realizadas sobre el sistema.

Arquitectura del Sistema

Se debe añadir al documento de requerimientos creado en el trabajo práctico anterior una sección que incluya un **diagrama de arquitectura preliminar** que permita asociar requerimientos y casos de usos con los sistemas, subsistemas y módulos identificados.

Por otro lado, se debe generar otro documento de **arquitectura** donde se presente:

- Un gráfico de arquitectura general para mostrar los componentes y sus relaciones con las interfaces externas.
- La explicación del **patrón de arquitectura** que fue usado y por qué, haciendo énfasis en cómo resuelve los requerimientos no funcionales.
- Diagramas de despliegue y de componentes.
- Opcionalmente se puede incluir un diagrama de contexto que incluya la relación de los sistemas y los subsistemas con las actividades del dominio del conocimiento de la aplicación.

Se deben definir los casos de **prueba de integración** que verifican la correcta interacción entre todos los componentes del sistema.

Diseño del Sistema

Se debe presentar un **documento de diseño** en el cual se incluyan diagramas de paquetes, diagramas de clases y objetos, diagramas de secuencia y/o todo aquel diagrama que sirva para explicar el diseño del software a construir.

Se debe mostrar también la aplicación de patrones de diseño usando diagramas adecuados que detallen su implementación e indicar por qué se utilizan y qué problemas se solucionan con ellos.

Por otro lado, se deben generar **pruebas unitarias automáticas para el código**, explicando cómo correrlas y verificar su estado.

Opcionalmente, se puede incluir cualquier otra herramienta de gestión de la calidad de software que haya sido usada. Mostrando ejemplos de su uso y se debe agregar al Plan de Gestión de las Configuraciones la dirección y forma de acceso a la herramienta.

Presentación Final

Además de todos los documentos generados en el Trabajo Práctico 1 y en los puntos anteriores del Trabajo Práctico 2, se deben presentar tres documentos:

- Notas de entrega del proyecto (release notes).
- Informe sobre el trabajo realizado.
- Presentación para mostrar ante sus compañeros.

La nota de entrega debe incluir:

- Breve listado de la funcionalidad incluida (con el estado de implementación de cada uno).
- “Pass/Fail Ratio (porcentaje de pruebas pasadas/falladas)” para todas las pruebas realizadas (pruebas de sistema, aceptación, integración y unitarias).
- Número de defectos identificados y corregidos (agrupados por severidad).
- Defectos conocidos (no resueltos) al momento de entregar el proyecto.
- Dirección de acceso o archivo con los archivos del proyecto y sus instrucciones de instalación y ejecución.

El informe sobre el trabajo realizado debe incluir:

- Detalle de dedicación de esfuerzo para realizar el trabajo (en personas horas), distinguiendo la contribución personal de cada miembro del grupo y el esfuerzo invertido por cada tarea realizada. Esto incluye todas las tareas para poder construir la aplicación, documentar y elaborar el informe y la presentación de esta materia.
- Lecciones aprendidas durante la elaboración del práctico y errores cometidos.

Las condiciones requeridas para los prácticos podrían ir actualizándose en función de las necesidades y las modernizaciones en las técnicas/metodologías.

Desagregado de competencias y resultados de aprendizaje

Las instancias de evaluación están desarrolladas para certificar que los estudiantes respalden de manera efectiva las competencias profesionales requeridas para el alcance de ésta materia. Las evaluaciones fueron alineadas a la metodología centrada en el estudiante y enfocadas en desarrollar habilidades prácticas.

Competencia	Resultado de Aprendizaje	Mínimo	Valoración
CG2 CE1.2	Interpreta correctamente el dominio de un problema.	2	
CG2 CG3	Detecta y comunica errores, fallas, defectos y oportunidades de mejoras tanto en el proceso	2	

CG7 CE1.2 CE4.3 CE4.4 CE8	como en el producto elaborado		
CG3 CG4 CE1.4 CE4.3 CE4.4 CE4.11	Selecciona y emplea adecuadamente las técnicas y herramientas a aplicar para un problema dado	2	
CG4 CG7 CE1.2 CE4.11 CE8	Comunica de Manera Efectiva los resultados de los procesos de implementaciones de herramientas, las mediciones, los inconvenientes, los desafíos y las acciones realizadas para superar los obstáculos	2	
CG3 CG7 CE1.2 CE4.11	Desarrolla habilidades comunicacionales efectivas para elicitar, redactar y validar requerimientos.	2	
CG8	Identifica los aspectos del desarrollo profesional que por su origen y naturaleza tienen connotaciones éticas	2	
CG3 CG7 CG8 CG10	Respeto las pautas de trabajo establecidas en clase para las actividades de equipo.	2	
CG3 CG7 CG10	Trabaja en equipo asumiendo los distintos roles dentro de un grupo de trabajo	2	
CG10 CE1.4 CE4.3 CE4.4	Lidera, Transforma, Motiva y/o Propone mejoras, cambios, adecuaciones y soluciones a los paradigmas actuales, a los procesos aprendidos, a los desarrollos actuales con el fin de encontrar propuestas superadoras que generen nuevas oportunidades.	2	

Se realizan autoevaluaciones periódicas buscando generar un espacio de reflexión sobre su progreso de aprendizaje y su desarrollo en dichas competencias. Esta autoreflexión les permite identificar oportunidades de mejoras y tomar decisiones para poder afrontar dicho desafío.

A los fines de evaluar los trabajos se emplearán rúbricas para que el estudiante pueda discernir los objetivos que alcanzó y en qué grado a modo de retroalimentación.

El rango de valoración es de 1 a 3 y se corresponde a:

1. Insuficiente: No se evidencia el nivel de desarrollo de las competencias esperado a través de los resultado de aprendizaje
2. Suficiente: En la mayoría de las situaciones se evidencia el nivel de desarrollo deseado.
3. Alto: Se evidencia un claro desarrollo de las competencias esperado a través de los resultados de aprendizaje.

Bibliografía

BIBLIOGRAFIA Principal

- Ian Sommerville (2016) Ingeniería del Software (10 Edición) Pearson

BIBLIOGRAFIA Complementaria

- Gergely Orosz (2021) Building Mobile Apps at Scale: 39 Engineering Challenges (2 Edición) Primedia E-launch LLC
- Eric Freeman (2021) Head First Design Patterns (2 Edición) O'Reilly Media
- Eric Evans (2003) Domain-Driven Design: Tackling Complexity in the Heart of Software (1 Edición) Pearson Education
- Robert C Martin (2017) Clean Architecture (1 Edición) Pearson
- Robert C Martin (2008) Clean Code (1 Edición) Pearson