

Programa Asignatura: Proyecto Informático

Unidad Académica Responsable: Departamento de Informática y Ciencias de la Computación

CARRERA a las que se imparte: Ingeniería Civil Informática

I.- IDENTIFICACIÓN

Nombre: Proyecto Informático		
Código: 503369	Créditos: 4	Créditos SCT: 6
Prerrequisitos: 503355 Ingeniería de Software II		
Modalidad: presencial	Calidad: obligatorio	Duración: semestral
Semestre en el plan de estudios: 10	Ingeniería Civil Informática	
Trabajo Académico: 10		
Horas Teóricas: 3	Horas Prácticas: 0	Horas Laboratorio: 2
Horas de otras actividades: 5		

II.- DESCRIPCIÓN

En esta asignatura, el alumno tendrá la oportunidad de integrar y materializar los conocimientos y competencias adquiridas en las asignaturas del plan de estudio y, en particular, de aquéllas del área de Ingeniería de Software, a través del desarrollo de un proyecto de Sistema Informático. El proyecto consiste en el desarrollo de un nuevo producto o servicio basado en la tecnología informática y su implementación a nivel de prototipo.

Paralelamente, se revisarán y analizarán contenidos que complementan el desarrollo del proyecto, con énfasis en la perspectiva de sistema informático como agente de innovación y cambio dentro de la organización.

En relación al perfil de egreso, la asignatura contribuye con las siguientes competencias:

- Aplicar principios de matemáticas, ciencias de la ingeniería y ciencias de la computación, a problemas de ingeniería informática.
- Identificar necesidades de usuarios, a partir de las cuales formula, diseña e implementa soluciones informáticas que satisfagan las especificaciones.
- Trabajar en equipo para solucionar problemas de manera colaborativa e incentivando el trabajo multidisciplinario.
- Comprender, desenvolverse y comunicarse efectivamente en los ambientes laborales delimitados por la responsabilidad profesional y ética.
- Comprender e identificar el impacto de soluciones informáticas en un contexto global, económico, ambiental y social.

III.- RESULTADOS DE APRENDIZAJE ESPERADOS

Al finalizar el curso los alumnos deben ser capaces de:

1. El alumno será capaz de formular un proyecto tecnológico considerando las etapas de: plan de negocio, especificaciones técnicas, diseño e implementación.
2. El alumno será capaz de comprender los aspectos involucrados en la generación de nuevas ideas, basadas en la tecnología informática.
3. El alumno será capaz de conocer y comprender conceptos asociados a los productos tecnológicos: generación, financiamiento y desarrollo.
4. El alumno será capaz de ejecutar todas las actividades involucradas en la formulación de un proyecto concreto.
5. El alumno será capaz de comprender aspectos contextuales de los proyectos informáticos: económicos, organizacionales y legales.
6. El alumno será capaz de formular y crear un proyecto en concreto.

IV.- CONTENIDOS

1. Emprendimiento y nuevas ideas para proyectos informáticos
 - Características del emprendedor
 - Barreras de Entrada
 - Inversionistas privados
 - Valor de la tecnología
 - Alianzas estratégicas, incubadoras y Spin-Off.
2. Proyecto Informático como instancia de Innovación y Cambio
 - Evolución y difusión de la innovación
 - Business Process Reengineering
 - Knowledge Management
 - Investigación y Desarrollo
 - Ciclos Tecnológicos
 - Diseño Dominante
 - Integración Tecnológica
3. Aspectos contextuales de proyectos informáticos
 - Consideraciones económicas
 - Aspectos organizacionales
 - Aspectos legales: patentes, copyright, trademarks, licencias
 - Marketing de productos tecnológicos
 - Consideraciones de producción

V.- METODOLOGÍA

El curso contará con clases expositivas y con trabajo colaborativo en el marco del desarrollo de un proyecto semestral. El alumno deberá tener una participación activa dentro del curso, desempeñando uno o más roles en el equipo de desarrollo encargado del proyecto semestral.

VI.- EVALUACIÓN

- Controles (escritos u orales) sobre la comprensión de contenidos relevantes de la asignatura.
- Evaluación basada en etapas y cumplimiento de metas en el desarrollo del proyecto semestral, tanto en el desempeño individual como grupal

VII.- BIBLIOGRAFÍA Y MATERIAL DE APOYO

Bibliografía Básica:

- James Cadle, Donald Yeates: Project Management for Information Systems, 5th edition. Prentice Hall, 2007. 464 pp., ISBN-10 0132068583, ISBN-13 978-0132068581.
- Richard C. Dorf: The Technology Management Handbook, CRC Press, 2000. 1184 pp., ISSN-10 0849385776, ISBN-13 978-0849385773.

Bibliografía Complementaria:

- Walker Royce: Software project management: a unified framework. Addison-Wesley, 1998. 448 pp., ISBN-10 0201309580, ISBN-13 978-0201309584.

Metodología de Trabajo semestre 2023-2

Profesores: Marcela Varas, Gonzalo Pérez

Sesiones sincrónicas:

Viernes de 10:15 a 12 (Sala Diseño Mecánico), y de 15:15 a 16:45 lab sistemas

Sesión asincrónica: Se recomienda usar bloque del jueves de 10 a 11

1. El curso será organizado en 8 equipos de entre 5 y 7 personas, y se asumirán los roles según lo aprendido en ingeniería de software 2 para el proyecto semestral.
2. Se desarrollará un proyecto semestral por equipo.
3. El proyecto a desarrollar obedece a un problema real de una comunidad de usuarios del campus UdeC o cercano geográficamente. Luego se debe proponer y validar una solución informática.
4. La identificación del problema y la solución se desarrollará utilizando Design Thinking.
5. El proyecto será planificado por cada equipo y reportado mes a mes de manera presencial, y de manera permanente en jira.
6. Las clases son obligatorias y se registrará asistencia.

Para el proyecto, cada calificación se obtiene del informe escrito, la presentación y la interrogación a cada uno de los miembros del equipo. La nota de la interrogación/presentación es la más baja del equipo, por lo cual se debe cautelar que todos participen y conozcan los detalles.

Cada informe se califica con un 50% de la versión escrita, y 50% de la presentación/interrogación.

Nota Final: Proyecto .6+ Evaluaciones Escritas*.4

Proyecto = Entrega 1 *.15 + entrega 2 *.35 + entrega 3 *.5

Se requiere tener nota >4 en cada informe del proyecto y en el promedio de las evaluaciones escrita. En caso contrario, no se cumple requisito.

Fecha	Actividad MVaras	Actividad GPerez
Vi 11/08 Semana 1 Definición equipo/pre selección Organización y problema	Presentación del Curso Asignación de Equipos Elección Problema/Organización Emprendimiento y nuevas ideas para proyectos informáticos. Código de ética y contrato equipo.	Asuntos clave para el desarrollo de un PI (intro). Estrategias para abordar proyectos (Conceptos claves) (Identificación, Diseño, Implementación y Eval)
Vi 18/08 Semana 2 Asignación de roles, planificación proyecto	Proceso de Innovación Innovación: Empatizar	Actividades de planificación Presentación del equipo/roles Selección de tecnologías para la planificación
Vi 25/08 Semana 3	Modelos de Negocio Definir	Reunión con el equipo para establecer los objetivos y alcance del proyecto. Identificación de los requisitos funcionales y no funcionales del software. Creación de un plan de proyecto detallado con fechas límite y responsabilidades claras. Establecimiento de una arquitectura inicial del software y elección de tecnologías clave.
Vi 1/09 Semana 4	Mapa de Empatía /Idear	Diseño de la arquitectura del software y la base de datos. Creación de prototipos de la interfaz de usuario para validar el diseño con los usuarios. Obtener retroalimentación de los interesados y realizar ajustes en el diseño según sea necesario.
Vi 8/09 Semana 5 Entrega 1	Entrega 1: presentación equipos	Desarrollo del software según las especificaciones y el diseño establecido. Enfoque en implementar las funcionalidades clave y los módulos principales. Pruebas unitarias para garantizar la calidad del código.
Vi 15/09 6	Rediseño de Procesos/Digitalización/Transformación Digital	Realizar una revisión del progreso del proyecto hasta el momento. Identificar posibles problemas y riesgos.

		Ajustar el plan si es necesario para asegurarse de que el proyecto esté en camino.
Vi 29/09 7	Transformación Digital	Realizar una revisión del progreso del proyecto hasta el momento. Identificar posibles problemas y riesgos. Ajustar el plan si es necesario para asegurarse de que el proyecto esté en camino.
Vi 6/10 8	Evaluación Escrita 1	Continuar con el desarrollo de las funcionalidades restantes. Integrar los módulos desarrollados hasta el momento. Realizar pruebas de integración para asegurar el correcto funcionamiento del sistema en conjunto.
Vi 13/10 9	Ecosistemas de Innovación	Continuar con el desarrollo de las funcionalidades restantes. Integrar los módulos desarrollados hasta el momento. Realizar pruebas de integración para asegurar el correcto funcionamiento del sistema en conjunto.
Vi 20/10 10 Entrega 2	Entrega 2: presentación e interrogación equipos	Realizar pruebas exhaustivas del software en diferentes escenarios. Identificar y corregir errores y fallos en el software.
Vi 27/10 (JCC2023) 11	Propiedad Intelectual	Realizar pruebas de rendimiento y optimización si es necesario. (Conclusiones de los alumnos de su propia arquitectura)
Vi 3/11 12	Evaluación Económica de TI	Preparar el entorno de producción para el lanzamiento del software. Capacitar a los usuarios finales y al personal relacionado con el sistema. (Si es requerido)
Vi 10/11 13	Evaluación Escrita 2	Lanzar el software y asegurarse de que todo funcione correctamente en el entorno de producción.

Vi 17/11 14	Entrega Final: Presentaciones Proyectos	Preparación de documentación final de la implementación (revisión de repositorios) Documentación de despliegue, manuales de usuario y capacitación, documentación de mantenimiento...
Vi 24/11 15 entrega final	Entrega Final: Presentaciones Proyectos	Entrega Final: Presentaciones Proyectos
Vi 1/12 16 entrega final	Entrega Final: Presentaciones Proyectos	Entrega Final: Presentaciones Proyectos
17 8/12 feriado/fin de clases		

Anexo 1- Requerimientos técnicos de la aplicación:

Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es una forma común de diseñar sistemas en los que diferentes componentes (clientes y servidores) interactúan para proporcionar funcionalidad y datos a los usuarios. Con tres componentes principales: la API REST, el front-end web y la aplicación móvil. También se puede combinar con enfoques de Microservicios, arquitectura de Micro Kernel y cualquier otra justificando su utilización.

API REST:

La API REST es el componente central de la arquitectura. Proporciona endpoints a través de los cuales tanto el front-end web como la aplicación móvil pueden solicitar y enviar datos. Debe diseñar tu API siguiendo las mejores prácticas RESTful, utilizando verbos HTTP adecuados (GET, POST, PUT, DELETE, etc.) para realizar operaciones en los recursos. Puedes considerar la autenticación y la autorización para garantizar la seguridad de la API.

FRONTEND WEB

El front-end web es la interfaz con la que los usuarios interactúan desde un navegador. Puede ser construido utilizando tecnologías como HTML, CSS y JavaScript, y puede utilizar frameworks como React, Angular o Vue.js para crear una experiencia de usuario interactiva. El front-end web se comunicará con la API REST para obtener y enviar datos utilizando solicitudes HTTP, generalmente a través de AJAX o Fetch.

APPMOVIL

La aplicación móvil es otra interfaz de usuario que los usuarios pueden utilizar desde sus dispositivos móviles. Puede desarrollarse utilizando lenguajes y frameworks específicos para plataformas como Swift para iOS o Kotlin para Android. La aplicación móvil también se comunicará con la API REST para obtener y enviar datos, al igual que el front-end web.

CONSIDERACIONES ADICIONALES

- Seguridad: Asegurar de implementar prácticas de seguridad sólidas en toda la arquitectura, incluida la autenticación y autorización adecuadas en la API REST.
- Escalabilidad: Diseñar la arquitectura de manera que pueda escalarse según sea necesario para manejar un mayor tráfico y usuarios.
- Cache: Considera la implementación de estrategias de caché para mejorar el rendimiento y reducir la carga en la API.
- Monitorización y registro: Implementar herramientas de monitorización y registro para supervisar el rendimiento y detectar posibles problemas en tiempo real.
- Actualizaciones: Asegurar de que los cambios en la API no afecten negativamente a las interfaces de usuario existentes.