|  | **PROGRAMA EDUCATIVO:**  **LICENCIATURA EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN E INNOVACIÓN DIGITAL**  **EN COMPETENCIAS PROFESIONALES** |  |
| --- | --- | --- |

**PROGRAMA DE ASIGNATURA:** \_\_**ANÁLISIS Y DISEÑO DE SOFTWARE**\_\_\_\_\_\_\_\_\_\_\_\_ **CLAVE:**\_\_\_\_**ADS**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Propósito de aprendizaje de la Asignatura | | El estudiante implementará metodologías y modelos de desarrollo a través del ciclo de vida del software, considerando el análisis, el diseño centrado en el usuario y la evaluación de la interfaz considerando los principios de la interacción humano - computadora, para dar solución a problemáticas planteadas. | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Competencia a la que contribuye la asignatura | | Desarrollar soluciones tecnológicas multiplataforma de software web y móvil utilizando programación orientada a objetos, frameworks, bases de datos, estándares de calidad y diseño para resolver problemas del sector productivo, con un enfoque de inclusión, compromiso con la responsabilidad social, equidad social y de género, excelencia, vanguardia, innovación social e interculturalidad. | | | | |
| Tipo de competencia | Cuatrimestre | | Créditos | Modalidad | Horas por semana | Horas Totales |
|
| **Específica** | **4** | | **4.6875** | **Escolarizada** | **5** | **75** |

| **Unidades de Aprendizaje** | **Horas del Saber** | **Horas del Saber Hacer** | **Horas Totales** |
| --- | --- | --- | --- |
|  |  |  |
| 1. Ingeniería de software | 5 | 20 | 25 |
| 1. Metodologías y Modelado de Software | 5 | 20 | 25 |
| 1. Diseño de interfaces | 5 | 20 | 25 |
| **Totales** | **15** | **60** | **75** |

| **Funciones** | **Capacidades** | **Criterios de Desempeño** |
| --- | --- | --- |
| Diseñar soluciones tecnológicas web y móviles con base en un análisis, utilizando algoritmos y estructuras de datos eficientes, patrones de diseño, base de datos, experiencia de usuario, metodologías y estándares con el fin de satisfacer los requerimientos considerados por la empresa. | •Realizar el análisis de los requerimientos del software utilizando estándares y técnicas de recopilación de información para el levantamiento, la validación y la especificación de los requerimientos que representen las necesidades del cliente. | Integrar un reporte técnico sobre el análisis de los requerimientos de software que documente:  - La identificación de flujos de datos del problema.  - Identificación de eventos(casos de uso).  - Identificar y determinar los requerimientos, tanto funcionales como no funcionales.  - Identificación del alcance.  - Validación de requisitos para asegurar su precisión, completitud y consistencia.  - Verificar que los requisitos sean factibles de implementar con las tecnologías disponibles.  - Identificar y gestionar los riesgos asociados a los requisitos. |
| Realizar el diseño funcional y arquitectónico de la aplicación utilizando herramientas, estructuras de datos y patrones de diseño para definir la representación física y lógica de la solución | Integrar un reporte técnico sobre el diseño del software utilizando diagramas UML que documente:  - La definición de las funcionalidades de la aplicación y su organización en módulos independientes.  - La descripción de los pasos que los usuarios seguirán para realizar las diferentes tareas dentro de la aplicación.  - La especificación de las reglas y lógica que gobiernan el comportamiento de la aplicación(Reglas del Negocio).  - Arquitectura y Patrones de diseño.  - La identificación de los componentes y servicios que conforman la aplicación y cómo interactúan entre sí.  - La selección de las tecnologías y frameworks que se utilizarán para desarrollar la aplicación.  - La selección de la estructura de datos adecuada para representar y organizar la información eficientemente en la aplicación. |
| Diseñar la interfaz de usuario de la aplicación utilizando estándares y criterios de usabilidad para mejorar la experiencia de usuario. | Integrar un reporte técnico sobre el diseño de las interfaces que incluya mockups con los componentes de diseño y control del software que documente:  - El diseño de la interfaz incluyendo pantallas, formularios, menús, etc.  - El diseño visual, incluyendo estética (uso de colores, tipografía e imágenes), consistencia y claridad.  - La interactividad, incluyendo la navegación, mensajes informativos al usuario.  - La accesibilidad, considerando la adaptabilidad a diferentes dispositivos y tamaños de pantalla, tecnologías de asistencia y legibilidad.  - La usabilidad, considerando simplicidad, eficiencia, ayuda y soporte. |

**UNIDADES DE APRENDIZAJE**

| Unidad de Aprendizaje | 1. Ingeniería de software | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Propósito esperado | El estudiante elaborará el documento de especificación de requerimientos de un proyecto de software para definir el comportamiento funcional del sistema. | | | | | |
| **Tiempo Asignado** | **Horas del Saber** | 5 | **Horas del Saber Hacer** | 20 | **Horas Totales** | 25 |

| **Temas** | **Saber**  **Dimensión Conceptual** | **Saber Hacer**  **Dimensión Actuacional** | **Ser y Convivir**  **Dimensión Socioafectiva** |
| --- | --- | --- | --- |
| Fundamentos de análisis de sistemas | Definir el concepto de análisis de software.  Describir el rol del analista de software.  Identificar los elementos del ciclo de vida del desarrollo de software.  Describir los tipos de software |  | - Desarrollar el razonamiento sistemático para ejecutar el proceso de desarrollo de software  - Desarrollar la Comunicación asertiva para detectar las necesidades de la organización y trabajar en equipo armoniosamente  - Trabajar de manera colaborativa para contribuir al logro de los objetivos del proyecto de software  - Desarrollar el pensamiento analítico a través de la identificación de conceptos para resolver problemas en su formación académica o su entorno.  - Asumir la responsabilidad y honestidad para realizar actividades en forma individual y en equipo de manera proactiva.  - Ejercer liderazgo, coordinando las actividades para el buen resultado del proceso a desarrollar.  - Impulsar la responsabilidad social para el desarrollo de proyectos inclusivos que contribuyan a mejorar la calidad de vida en la sociedad |
| Tipos de sistemas de información | Identificar los tipos de sistemas de información |  |
| Fundamentos de procesos organizacionales en el desarrollo de software | Definir el concepto de proceso organizacional  Identificar los principales procesos de una organización  Describir las características de los procesos organizacionales |  |
| Técnicas de recolección de datos y requerimientos | Definir el concepto de análisis de requerimientos  Describir los tipos de requerimientos de software  Distinguir los requerimientos de software de acuerdo al estándar IEEE 830-1998, utilizando técnicas de validación de requerimientos  Plantillas SRS | Seleccionar el método de recopilación de información más adecuado para el tipo de proyecto de software como: guía de entrevista, encuesta, guía de observación y lista de verificación.  Diseñar los instrumentos para la recolección de datos |
| Especificación y validación de requerimientos. | Definir el concepto de metodología tradicional de desarrollo  Identificar las fases de las metodologías tradicionales de desarrollo  Distinguir las ventajas y desventajas de las metodologías tradicionales de desarrollo. | Clasificar los requerimientos para un proyecto de software.  Redactar los requerimientos funcionales y no funcionales para un proyecto de software  Proponer la plantilla adecuada para el tipo de proyecto de acuerdo al estándar IEEE830. |

| **Proceso Enseñanza-Aprendizaje** | | | |
| --- | --- | --- | --- |
| Métodos y técnicas de enseñanza | Medios y materiales didácticos | Espacio Formativo | |
| **Aula** |  |
| - Equipos colaborativos.  - Tareas de Investigación  - Solución de problemas. | Pizarrón.  Plumones.  Computadora.  Internet.  Equipo multimedia.  Ejercicios prácticos.  Plataformas virtuales. | **Laboratorio / Taller** | X |
|  |  | **Empresa** |  |

| **Proceso de Evaluación** | | |
| --- | --- | --- |
| Resultado de Aprendizaje | Evidencia de Aprendizaje | Instrumentos de evaluación |
| Los estudiantes establecen los requerimientos para definir el comportamiento y tipo de sistema de acuerdo con el estándar IEEE 830 con base a los fundamentos de análisis de sistemas y la comprensión de los procesos organizacionales aplicando técnicas de recolección de datos. | A partir de un caso práctico, elaborar un reporte digital que contenga la "Especificación de Requerimientos de Software" (ERS) que incluya:  • Fecha  • Nombre del Proyecto  • Resumen  • Introducción  - Antecedentes del cliente.  - Estructura de la organización.  - Justificación social del proyecto con base a criterios internacionales (macrotendencias, ODS)  • Desarrollo  - Objetivo  - Alcance  - Justificación de las técnicas de recolección de datos elegidas.  - Requerimientos:  • Funcionales  • No funcionales  - Lista de cotejo de la revisión de requisitos  • Conclusiones. | Estudios de casos  Rúbrica |

| Unidad de Aprendizaje | 1. Metodologías y Modelado de Software | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Propósito esperado | El estudiante diseñará el modelo de un proyecto de software aplicando las metodologías de desarrollo para definir las características del sistema. | | | | | |
| **Tiempo Asignado** | **Horas del Saber** | 5 | **Horas del Saber Hacer** | 20 | **Horas Totales** | 20 |

| **Temas** | **Saber**  **Dimensión Conceptual** | **Saber Hacer**  **Dimensión Actuacional** | **Ser y Convivir**  **Dimensión Socioafectiva** |
| --- | --- | --- | --- |
| Metodologías tradicionales | Definir el concepto de metodología tradicional de desarrollo  Identificar las fases de las metodologías tradicionales de desarrollo  Distinguir las ventajas y desventajas de las metodologías tradicionales de desarrollo. | Seleccionar la metodología tradicional apropiada de acuerdo al tipo de proyecto. | - Desarrollar el razonamiento sistemático para ejecutar el proceso de desarrollo de software  - Desarrollar la Comunicación asertiva para trabajar en equipo armoniosamente  - Trabajar de manera colaborativa para contribuir al logro de los objetivos del proyecto de software  - Desarrollar el pensamiento analítico a través de la identificación de | para resolver problemas en su formación académica o su entorno  - Asumir la responsabilidad y honestidad para realizar actividades en forma individual y en equipo de manera proactiva.  - Ejercer liderazgo, coordinando las actividades para el buen resultado del proceso a desarrollar.  - Organizar y planificar los componentes de un modelo de software para establecer sus principales funcionalidades  - Impulsar la responsabilidad social para el desarrollo de proyectos inclusivos que contribuyan a mejorar la calidad de vida en la sociedad |
| Metodologías ágiles | Definir el concepto de metodología de desarrollo ágil  Identificar las fases de las metodologías ágiles de desarrollo  Distinguir las ventajas y desventajas de las metodologías de desarrollo ágiles. | Seleccionar la metodología ágil que se adapte a las condiciones de un proyecto de software. |
| Modelado UML | Identificar la estructura del lenguaje UML de acuerdo a las áreas estructural, dinámica, gestión del modelo y extensiones.  Describir los diagramas de modelado UML: Casos de uso, diagrama de clases, Diagramas de actividades, secuencia, componentes, despliegue.  Explicar el proceso de construcción de diagramas en el modelado de requerimientos: comportamiento, casos de uso, procedimientos y actividades. | Seleccionar la metodología tradicional apropiada de acuerdo al tipo de proyecto.  Seleccionar la metodología ágil que se adapte a las condiciones de un proyecto de software.  "Diseñar el modelado de software mediante la estructura estática y dinámica de UML (Casos de uso, clases, secuencia, componentes, despliegue, estado).  Realizar el proceso de normalización de los diagramas del modelado de requerimientos.  Verificar el cumplimiento de construcción de los diagramas UML según el modelo de negocio. |

| **Proceso Enseñanza-Aprendizaje** | | | |
| --- | --- | --- | --- |
| Métodos y técnicas de enseñanza | Medios y materiales didácticos | Espacio Formativo | |
| **Aula** |  |
| - Equipos colaborativos.  - Tareas de Investigación  - Solución de problemas. | Pizarrón.  Plumones.  Computadora.  Internet.  Equipo multimedia.  Ejercicios prácticos.  Plataformas virtuales. | **Laboratorio / Taller** | X |
|  |  | **Empresa** |  |

| **Proceso de Evaluación** | | |
| --- | --- | --- |
| Resultado de Aprendizaje | Evidencia de Aprendizaje | Instrumentos de evaluación |
| Los estudiantes diseñan el modelo del proyecto de software en lenguaje UML y seleccionan la metodología de desarrollo adecuada de acuerdo con las características del mismo. | A partir de un caso práctico, elaborar un reporte digital que contenga el modelado del software incluyendo lo siguiente:  • Selección de la metodología de desarrollo para el proyecto de software  • Justificación de la selección de la metodología de desarrollo apropiada para el proyecto de software  • Casos de uso.  • Clases.  • Secuencia.  • Colaboración.  • Estado. | Ejercicios prácticos.  Rúbrica |

| Unidad de Aprendizaje | 1. Diseño de interfaces | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Propósito esperado | El estudiante implementará principios de interacción humano - computadora, evaluando la usabilidad en un modelo de diseño centrado en el usuario para la elaboración de prototipos de interfaces. | | | | | |
| **Tiempo Asignado** | **Horas del Saber** | 5 | **Horas del Saber Hacer** | 20 | **Horas Totales** | 25 |

| **Temas** | **Saber**  **Dimensión Conceptual** | **Saber Hacer**  **Dimensión Actuacional** | **Ser y Convivir**  **Dimensión Socioafectiva** |
| --- | --- | --- | --- |
| Criterios de usabilidad | Definir el concepto de usabilidad.  Explicar los atributos de usabilidad de software. |  | - Desarrollar el razonamiento sistemático para ejecutar el proceso de desarrollo de software  - Desarrollar la Comunicación asertiva para trabajar en equipo armoniosamente  - Trabajar de manera colaborativa para contribuir al logro de los objetivos del proyecto de software  - Desarrollar el pensamiento analítico a través de la identificación de conceptos para resolver problemas en su formación académica o su entorno  - Asumir la responsabilidad y honestidad para realizar actividades en forma individual y en equipo de manera proactiva.  - Ejercer liderazgo en la práctica de laboratorio, coordinando las actividades para el buen resultado de la práctica o proceso a desarrollar  - Organizar y planificar el proceso de diseño de interfaces y evaluación de la usabilidad de un proyecto de software  - Impulsar la responsabilidad social para el desarrollo de proyectos inclusivos que contribuyan a mejorar la calidad de vida en la sociedad |
| Explicar el proceso de diseño de usabilidad del software. |  |
| Definir el concepto de accesibilidad. |  |
| Describir el modelo de diseño universal. |  |
| Describir el proceso de aseguramiento de la accesibilidad de los sistemas. | Proponer mejoras de usabilidad del software de acuerdo a su expectativa de utilización. |
| Describir el proceso de caracterización de las capacidades humanas diferentes. | Caracterizar las capacidades humanas diferentes. |
| Experiencia de usuario | Definir el concepto de experiencia del usuario (UX) e interfaz de usuario (UI).  Identificar las diferencias entre UI y UX. | Diseñar interfaces de usuario (IU) centrado en la experiencia del usuario (UX). |
| Definir el proceso de experiencia de usuario: Investigación, organización, diseño, prototipos, pruebas y evaluación. | Elegir metodologías y técnicas de diseño centradas en el usuario. |
| Definir las características UX y los factores críticos: útil, usable, deseable, encontrable, accesible, creíble. | Documentar las características de los usuarios: perfil, habilidades, edad, educación, ubicación geográfica, plataforma que utiliza. |
| Describir las características y especificaciones de las UI. | Planear el proceso de experiencia del usuario. |
| Describir los procesos de diseño centrado en el usuario:  - Análisis centrado en el usuario.  - Ciclo de vida de la IU.  - Análisis de Tareas. | Determinar herramientas de diseño de interfaces. |
| Identificar el concepto de mockups y wireframes. | Diseñar prototipos basados en mockups. |
| Explicar el proceso de diseño e integración de técnicas y herramientas visuales en las UI con características UX. | Diseñar los wireframes de interfaces. |
| Definir el concepto de guía de estilo  Identificar los elementos que integran una guía de estilo | Seleccionar guías de estilo visual |
| Diseño de la interacción hombre-máquina | Definir el concepto de Interacción Humano-Computadora. |  |
| Identificar el concepto de modelos de navegación. |  |
| Describir los estilos y paradigmas de interacción. |  |
| Describir el proceso de identificación de elementos de hardware y software de la interacción con sistemas informáticos. | Interpretar los elementos de una interfaz de usuario. |
| Describir las características de la interacción con periféricos e interfaces de hardware. | Seleccionar dispositivos de interacción humano - computadora de acuerdo con las características del proyecto |
| Identificar los mecanismos de comunicación en la interacción humano – computadora. | Establecer los mecanismos de comunicación de acuerdo con las especificaciones para la interacción humano - computadora |
| Evaluación de la usabilidad e interfaces UX. | Identificar las técnicas y herramientas de evaluación formativa de interfaces (análisis comparativo A/B, evaluación heurística, think aloud, cognitive walkthrough, cuestionarios, clickstreams, eye-tracking.) | Determinar la herramienta para evaluación formativa.| |  |
| Identificar las técnicas de evaluación sumativa de interfaces. | Evaluar los prototipos de las interfaces de usuario. |
| Identificar los elementos que integran un experimento de usuario. | Determinar la herramienta de evaluación de accesibilidad. |
| Identificar las técnicas y herramientas de evaluación de accesibilidad de interfaces. | Evaluar la accesibilidad de las interfaces del producto desarrollado.  Evaluar el producto desarrollado. |

| **Proceso Enseñanza-Aprendizaje** | | | |
| --- | --- | --- | --- |
| Métodos y técnicas de enseñanza | Medios y materiales didácticos | Espacio Formativo | |
| **Aula** |  |
| - Equipos colaborativos.  - Tareas de Investigación  - Solución de problemas. | Pizarrón.  Plumones.  Computadora.  Internet.  Equipo multimedia.  Ejercicios prácticos.  Plataformas virtuales | **Laboratorio / Taller** | X |
|  |  | **Empresa** |  |

| **Proceso de Evaluación** | | |
| --- | --- | --- |
| Resultado de Aprendizaje | Evidencia de Aprendizaje | Instrumentos de evaluación |
| Los estudiantes construyen el prototipo de interfaces centrado en el usuario, considerando los principios de interacción humano - computadora, evaluando la usabilidad y la accesibilidad de un proyecto de software | Elabora un reporte digital a partir de un caso práctico, que incluya:  - Resumen  - Introducción  - Desarrollo:  - Documentación del proceso de diseño.  - Diseño de prototipo: Mockups, wireframes y modelo de navegación.  - Diseño de interfaces de usuario.  - Justificación de estilo de interacción.  - Descripción de los mecanismos de accesibilidad orientados al usuario  - Descripción de los atributos que mejoran la usabilidad del software  - Listado de dispositivos de interacción humano-computadora.  - Justificación de la guía de estilo utilizada.  - La guía de estilo.  - Definición del protocolo de evaluación formativa  - Los resultados de la evaluación formativa de los prototipos de interfaces de usuario.  - Definición del protocolo de evaluación sumativa.  - Los resultados de la evaluación sumativa del producto desarrollado.  - Definición del protocolo de evaluación de accesibilidad  - Los resultados de la evaluación de accesibilidad de las interfaces del producto desarrollado.  - Conclusiones  - Resultado general del proyecto  - Aporte del proyecto hacia el entorno social  - Referencias | Caso de estudio.  Rúbrica |

| **Perfil idóneo del docente** | | |
| --- | --- | --- |
| **Formación académica** | **Formación Pedagógica** | **Experiencia Profesional** |
| Ing. en Desarrollo y Gestión de software, Ing. Tecnologías de la información, Ing. Software, Ing. Sistemas Computacionales, Ing. Informática o carrera afín | Manejo de estrategias didácticas para enseñanza-aprendizaje, herramientas de evaluación, plataformas tecnológicas, técnicas de manejo de grupos, diseño de planeaciones didácticas entre otros | Analista de sistemas, Ingeniero de Software, Gestor de Proyectos de Desarrollo de Software, Arquitecto de Software, Analista de Requerimientos de Software, Desarrollador de Interfaces de Usuario (UI/UX), Consultor en Metodologías Ágiles, Programador, Docente con experiencia en el área, o afín. |

| **Referencias bibliográficas** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Autor | Año | Título del documento | Lugar de publicación | Editorial | ISBN |
| Kenneth C. Laudon, Jane P. Laudon | 2012 | 12a. Edición  Sistemas de Información Gerencial | México | Ed. Pearson | 978-607-32-0949-6 |
| Carmen de Pablos Heredero, José Joaquín López Hermoso Agius, Santiago Martín-Romo Romero, Sonia Medina Salgado | 2019 | Organización y transformación de los sistemas de información en la empresa | España | ESIC Editorial | 9788417513740, 8417513744 |
| Joyanes, Luis | 2015 | 6a. Edición - Sistemas de Información en la Empresa | México | Ed. Alfaomega | 9789587780024 |
| Sebastián Rubén Gómez Palomo, Eduardo Moraleda Gil | 2020 | Aproximación a la ingeniería del software | España | Editorial Universitaria Ramón Areces | 9788499613291, 8499613292 |
| Roger S. Pressman | 2010 | 7a. Edición -  Ingeniería de Software | México | Mc Graw Hill | 978-607-15-0314-5 |
| Ian Sommerville | 2016 | 9na Edición  Ingeniería de Software | España | Pearson | 978-607-32-0603-7 |
| Eric J. Braude, Michael E. Bernstein. | 2016 | Software Engineering: Modern Approaches, Second Edition. | USA | Waveland Press. | 978-1-4786-3230-6 |
| Martin Fowler | 2018 | 3rd edition  UML Distilled: A Brief Guide to the Standard Object Modeling Language | USA | Pearson Education | 9780134865126, 013486512X |
| Howard Podeswa | 2010 | 2nd Edición  UML For The IT Business Analyst | USA | Course Technology/Cengage Learning | 9781598638684, 1598638688 |
| Craig Larman | 2003 | Uml y Patrones  Una Introducción Análisis Y Diseño Orientado A Objetos Y Al Proceso Unificado | España | Pearson Educación | 9788420534381, 8420534382 |
| Donald A. Norman | 2013 | Segunda edición 1976  The Design of Everyday Things | USA | Basic Books | 978-0-465-07299-6 |
| Jeff Gothelf, Josh Seiden | 2021 | 3nd Edition  Lean UX: Designing Great Products with Agile Teams | USA | O'Reilly Media | 9781098116279, 1098116275 |
| Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven M. Jacobs, Niklas Elmqvist | 2018 | Designing the User Interface: Strategies for Effective Human-Computer Interaction. | USA | Pearson | 9781292153919, 1292153911 |

| **Referencias digitales** | | | |
| --- | --- | --- | --- |
| Autor | Fecha de recuperación | Título del documento | Vínculo |
| ACM. | 24/04/2024. | Software Engineering Code of Ethics. | http://www.acm.org/about/se-code-s |
| IEEE | 24/04/2024. | Software and Systems Engineering Standards. | https://standards.ieee.org/findstds/standard/software\_and\_systems\_engineering.html |
| Microsoft | 24/04/2024 | La guía sencilla para la diagramación de UML y el modelado de la base de datos | https://www.microsoft.com/es-es/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling |
| Ana Milagro Luzardo Alliey. | 24/04/2024 | Diseño de la interfaz gráfica web en función de los dispositivos móviles caso de estudio: diarios digitales. | http://www.palermo.edu/dyc/maestria\_diseno/pdf/tesis.completas/43.luzardo.pdf |
| Universidad Málaga | 24/04/2024 | Ejemplo de Guía de estilo | https://www.uma.es/media/files/GUIA\_WEB.pdf |
|  |  |  |  |