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

**PROGRAMA DE ASIGNATURA:** \_\_**PROGRAMACIÓN ORIENTADA A OBJETOS**\_\_\_\_\_\_\_\_ **CLAVE:­­­**\_\_\_\_**POO**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Propósito de aprendizaje de la Asignatura | El estudiante analizará, diseñará y codificará algoritmos computacionales a través del paradigma de programación orientado a objetos para resolver problemas del ámbito productivo e implementar soluciones eficientes. |
| Competencia a la que contribuye la asignatura | Desarrollar soluciones tecnológicas a través de lenguajes de programación estructurada, programación orientada a objetos y de consulta, herramientas de desarrollo asistido de software, usabilidad y pruebas, fundamentos de redes de área local, sistemas operativos, medidas de seguridad informática para contribuir a la eficiencia y productividad en diferentes contextos con un enfoque de impulso al desarrollo social, ambiental y de economía socialmente responsable. |
|  Tipo de competencia | Cuatrimestre | Créditos | Modalidad | Horas por semana | Horas Totales |
|
| **Específica** | **3** | **6.5625** | **Escolarizada** | **7** | **105** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unidades de Aprendizaje** | **Horas del Saber** | **Horas del Saber Hacer** | **Horas Totales** |
|  |  |  |
| 1. Modelado
 | 6 | 8 | 14 |
| 1. Sintaxis y documentación del Lenguaje Orientado a Objetos
 | 11 | 17 | 28 |
| 1. Gestión de Datos y buenas prácticas de codificación
 | 11 | 17 | 28 |
| 1. Entornos de desarrollo y diseño de aplicaciones
 | 14 | 21 | 35 |
| **Totales** | **42** | **63** | **105** |

|  |  |  |
| --- | --- | --- |
| **Funciones** | **Capacidades** | **Criterios de Desempeño** |
| Implementar soluciones básicas de software utilizando lenguajes de programación estructurada y orientada a objetos, aplicando herramientas básicas de desarrollo de software como entornos de desarrollo para contribuir a satisfacer las necesidades de la organización. | Diseñar aplicaciones básicas de software utilizando algoritmos, diagramas de flujo y casos de uso para la representación de la lógica de negocio, aplicando principios básicos de diseño funcional y seleccionando lenguajes de programación y herramientas de desarrollo de software adecuados a las necesidades y requerimientos del proyecto. | Diseña diagramas funcionales que representen la lógica de negocio de una aplicación básica, considerando: algoritmos, diagramas de flujo y casos de uso.Propone interfaces de usuario con elementos básicos de usabilidad. Utiliza lenguajes de programación y herramientas de desarrollo de software de acuerdo a las necesidades del proyecto. |
| Codificar aplicaciones básicas de software utilizando lenguajes de programación estructurada y orientada a objetos, empleando herramientas básicas de desarrollo de software en diversos entornos de desarrollo. | Codifica aplicaciones básicas de software utilizando lenguajes de programación estructurada y orientada a objetos a través de un código documentado con las siguientes características:- Integración del diseño de la aplicación: algoritmo, diagrama de flujo y casos de uso.- Utilizando estándares y técnicas de codificación y documentación. |
| Evaluar aplicaciones básicas de software aplicando pruebas para la detección y corrección de errores para asegurar su correcto funcionamiento. | Ejecuta pruebas de software para detectar y corregir errores.Documenta los resultados de las pruebas.Asegura el cumplimiento de los criterios de éxito con base en los requerimientos. |

**UNIDADES DE APRENDIZAJE**

|  |  |
| --- | --- |
| Unidad de Aprendizaje | 1. Modelado
 |
| Propósito esperado  | El estudiante determinará los requisitos y requerimientos mediante el uso del lenguaje unificado de modelado UML para desarrollar una solución que satisfaga las necesidades específicas de una situación a resolver. |
| **Tiempo Asignado** | **Horas del Saber**  | 6 | **Horas del Saber Hacer** | 8 | **Horas Totales** | 14 |

| **Temas** | **Saber****Dimensión Conceptual** | **Saber Hacer****Dimensión Actuacional** | **Ser y Convivir****Dimensión Socioafectiva** |
| --- | --- | --- | --- |
| Introducción a los conceptos del paradigma orientado a objetos (clases y objetos, atributos y métodos, relaciones estáticas y dinámicas entre objetos, herencia, polimorfismo, encapsulamiento e instanciamiento de objetos) | Identificar el concepto de paradigma de programación orientado a objetos.Identificar las características del paradigma orientado a objetos como es:-Clases y objetos-Atributos y métodos-Instancia-Abstracción-Encapsulamiento-Constructor-Herencia-Polimorfismo | Realizar la interpretación de un objeto.Definir sus atributos y métodos que llevará el objeto.Construir objetos a partir de una clase.Determinar solamente los atributos necesarios para la situación a resolver.Realizar la optimización por medio de herencia y polimorfismo. | Desarrollar el pensamiento analítico a través de la identificación de conceptos para resolver problemas en su formación académica y su entorno.Asumir la responsabilidad y honestidad para realizar actividades en forma individual y en equipo en forma proactiva.Incentivar la creatividad al explorar diferentes enfoques para la solución de problemas algorítmicos, valorando la diversidad de ideas y perspectivas. |
| Concepto de casos de uso, elementos de caso de uso (actor, objetivo, flujo de eventos). | Definir el concepto de modelado y diagrama en el paradigma de programación orientado a objetos. | Construir diagramas del lenguaje de modelado unificado |
| Modelado UML | Identificar los componentes del lenguaje de modelado unificado. | Determinar las funcionalidades primordiales que tendrá el proyecto |
| Patrones básicos de diseño (singleton, factory, observer) | Describir bloques de construcción del lenguaje de modelado unificado como:-Elementos-Relaciones-DiagramasExplicar el proceso de análisis del lenguaje de modelado unificado.Describir las características de los diagramas del lenguaje de modelado unificado:-Clases.-Objetos. -Componentes.-Despliegue. -Casos de uso.-Secuencia.-Colaboración.-Estados.-Actividades. | Determinar los actores que intervendrán en el proyectoConstruir los diagramas necesarios para la solución del problema. |

|  |
| --- |
| **Proceso Enseñanza-Aprendizaje** |
| Métodos y técnicas de enseñanza | Medios y materiales didácticos | Espacio Formativo |
| **Aula** | X |
| Aula invertidaInstrucción programadaResolver situaciones problemáticas | ProyectorPizarrónBibliografíaBuscadores académicosLecciones en un LMSAcceso a internetSoftware de desarrollo | **Laboratorio / Taller** |  |
|  |  | **Empresa** |  |

|  |
| --- |
| **Proceso de Evaluación** |
| Resultado de Aprendizaje | Evidencia de Aprendizaje | Instrumentos de evaluación |
| Los estudiantes identifican requisitos y requerimientos mediante el uso del lenguaje unificado de modelado UML desarrollando soluciones que satisfagan las necesidades específicas de una situación. | A partir de un portafolio de evidencias de prácticas identificar requisitos y requerimientos mediante el uso del lenguaje unificado de modelado UML desarrollando soluciones que satisfagan las necesidades específicas de situaciones planteadas. | Ejercicios prácticosLista de verificación |

|  |  |
| --- | --- |
| Unidad de Aprendizaje | 1. Sintaxis y documentación del Lenguaje Orientado a Objetos
 |
| Propósito esperado  | El estudiante representará clases en lenguaje de programación orientado a objetos para codificar el análisis y diseño de modelos en UML. |
| **Tiempo Asignado** | **Horas del Saber**  | 11 | **Horas del Saber Hacer** | 17 | **Horas Totales** | 28 |

| **Temas** | **Saber****Dimensión Conceptual** | **Saber Hacer****Dimensión Actuacional** | **Ser y Convivir****Dimensión Socioafectiva** |
| --- | --- | --- | --- |
| Sintaxis de un lenguaje de programación orientada a objetos. | Conocer la sintaxis que se utiliza en algún lenguaje de programación a objetos | Construir la sintaxis orientada a objetos en algún lenguaje de desarrollo | Fomentar la colaboración y la comunicación efectiva en equipos de desarrollo de soluciones computacionales, reconociendo la importancia del trabajo en conjunto para alcanzar soluciones funcionales.Desarrollar el pensamiento analítico a través de la identificación de conceptos para resolver problemas en su formación académica y su entorno.Incentivar la creatividad al explorar diferentes enfoques para la solución de problemas algorítmicos, valorando la diversidad de ideas y perspectivas. |
| Programación de objetos gráficos. | Identificar métodos y funciones simples y con parámetros en la sintaxis que conforma en el lenguaje de programación orientado a objetos | Construir funciones simplesConstruir funciones con parámetrosConstruir métodos simplesConstruir métodos con parámetros |
| Eventos en objetos gráficos. | Definir el concepto de interfaz gráfica de usuario en la programación orientada a objetos.Describir las características de los componentes de interfaz gráfica de usuarioIdentificar la jerarquía de los componentes gráficos Identificar las herramientas que se pueden utilizar en las aplicaciones del lenguaje de programación orientado a objetos. | Codificar componentes de interfaces gráficasDeterminar eventos en los componentes de interfaces gráficasDeterminar herramientas en tiempo de ejecución |
| Clases y objetos:- Atributos, métodos, relaciones estáticas y dinámicas entre objetos.- Herencia y polimorfismo.- Encapsulamiento e instanciamiento de objetos. | Definir las diferencias de:-Clases-Objetos-Atributos-MétodosDefinir las características de:-Herencia-Polimorfismo-Encapsulamiento-Instancia | Codificar clases y objetosAñadir atributos y métodos a las clasesProponer herencia para reducir atributosConstruir polimorfismo en las clases para optimizar el códigoProteger los atributos, métodos y funciones por medio del encapsulamiento |

|  |
| --- |
| **Proceso Enseñanza-Aprendizaje** |
| Métodos y técnicas de enseñanza | Medios y materiales didácticos | Espacio Formativo |
| **Aula** |  |
| Equipos cooperativosInstrucción programadaResolver situaciones problemáticas | ProyectorPizarrónBibliografíaBuscadores académicosLecciones en un LMSAcceso a internetSoftware de desarrollo | **Laboratorio / Taller** | X |
|  |  | **Empresa** |  |

|  |
| --- |
| **Proceso de Evaluación** |
| Resultado de Aprendizaje | Evidencia de Aprendizaje | Instrumentos de evaluación |
| Los estudiantes comprenden las clases en lenguaje de programación orientado a objetos en el análisis y diseño de modelos en UML. | A partir de un portafolio de evidencias de prácticas realizar programas que incluyan clases en lenguaje de programación orientado a objetos incluyendo el análisis y diseño de modelos en UML. | Ejercicios prácticosLista de verificación |

|  |  |
| --- | --- |
| Unidad de Aprendizaje | 1. Metodologías y herramientas del desarrollo del programa
 |
| Propósito esperado  | El estudiante desarrollará aplicaciones por medio del paradigma orientado a objetos con almacenamiento persistente en los datos. |
| **Tiempo Asignado** | **Horas del Saber**  | 11 | **Horas del Saber Hacer** | 17 | **Horas Totales** | 28 |

| **Temas** | **Saber****Dimensión Conceptual** | **Saber Hacer****Dimensión Actuacional** | **Ser y Convivir****Dimensión Socioafectiva** |
| --- | --- | --- | --- |
| Adquisición, procesamiento y almacenamiento de datos.- Conexión y consulta de base de datos | Definir el concepto de conexión a base de datos.Describir las características de conexión a base de datos.-Conectores de base de datos.Describir el proceso de conexión a base de datos. | Construir conexión a base de datos en clases y métodos con programación orientada a objetos.Construir consultas SQL en el lenguaje de programación. | Asumir la ética y responsabilidad de forma individual en las acciones de su entorno.Fomentar la colaboración y la comunicación efectiva en equipos de desarrollo de soluciones computacionales, reconociendo la importancia del trabajo en conjunto para alcanzar soluciones funcionales.Desarrollar el pensamiento analítico a través de la identificación de conceptos para resolver problemas en su formación académica y su entorno. |
| Documentación de código.- Documentación de la aplicación (manual de usuario, guía de instalación) | Describir sobre la documentación que se llevan en las aplicaciones.Definir los conceptos de:-Manual de usuario-Guía de instalación | Documentar el manual de usuarioDocumentar la guía de instalación con los requerimientos |
| Buenas prácticas de codificación (comentarios, indentación, nombres significativos) (clean code).- Principios de SOLID | Identificar las buenas prácticas que se llevan en la programación:-Comentarios-IndotaciónDefinir los principios de SOLID | Programar aplicaciones con el uso de buenas prácticas.Construir con base en los principios SOLID aplicaciones de programación orientada a objetos. |

|  |
| --- |
| **Proceso Enseñanza-Aprendizaje** |
| Métodos y técnicas de enseñanza | Medios y materiales didácticos | Espacio Formativo |
| **Aula** |  |
| Equipos cooperativosInstrucción programadaResolver situaciones problemáticas | ProyectorPizarrónBibliografíaBuscadores académicosLecciones en un LMSAcceso a internetSoftware de desarrollo | **Laboratorio / Taller** | X |
|  |  | **Empresa** |  |

|  |
| --- |
| **Proceso de Evaluación** |
| Resultado de Aprendizaje | Evidencia de Aprendizaje | Instrumentos de evaluación |
| Los estudiantes comprenden y analizan aplicaciones por medio del paradigma orientado a objetos con almacenamiento persistente en los datos. | A partir de un portafolio de evidencias de prácticas realizar aplicaciones por medio del paradigma orientado a objetos que incorporen el almacenamiento persistente en los datos. | Ejercicios prácticosLista de verificación |

|  |  |
| --- | --- |
| Unidad de Aprendizaje | IV. Entornos de desarrollo y diseño de aplicaciones |
| Propósito esperado  | El estudiante implementará aplicaciones con interfaz gráfica de usuario y con conexión a base de datos para resolver problemas del mundo real. |
| **Tiempo Asignado** | **Horas del Saber**  | 14 | **Horas del Saber Hacer** | 21 | **Horas Totales** | 35 |

| **Temas** | **Saber****Dimensión Conceptual** | **Saber Hacer****Dimensión Actuacional** | **Ser y Convivir****Dimensión Socioafectiva** |
| --- | --- | --- | --- |
| Entornos de desarrollo integrados:- Editores de código- Herramientas de depuración.- Control de versiones | Describir las características del entorno de desarrollo de los lenguajes de programación orientado a objetos:- Editor de código fuente.- Control de ejecución del programa.- Control de compilación.- Control de revisión de sintaxis.- Control del modo depuración. | Proponer el entorno de desarrollo del paradigma de programación orientado a objetos. | Asumir la ética y responsabilidad de forma individual en las acciones de su entorno.Fomentar la colaboración y la comunicación efectiva en equipos de desarrollo de soluciones computacionales, reconociendo la importancia del trabajo en conjunto para alcanzar soluciones funcionales.Desarrollar el pensamiento analítico a través de la identificación de conceptos para resolver problemas en su formación académica y su entorno.Incentivar la creatividad al explorar diferentes enfoques para la solución de problemas algorítmicos, valorando la diversidad de ideas y perspectivas. |
| Diseño de Aplicaciones Orientadas a Objetos:-Implementación de un sistema utilizando los principios Orientados a Objetos-Integración de aplicaciones con conexión a base de datos.-Pruebas Unitarias-Control de Excepciones-Control de Versiones (multiusuarios) | Definir las reglas de conexión para conectar aplicaciones con base de datosDefinir los conceptos de error y excepción.Describir las características de errores y excepciones en la ejecución de programa.Identificar los tipos de errores.Identificar las jerarquías de excepciones.Identificar los tipos de excepciones:-Implícitas.-Explícitas.Explicar la relación de tipo de errores y excepciones que se presentan en la ejecución de programa.Identificar los mecanismos de manejo de excepciones:-Bloques de manejo.-Propagación.-Captura.-Análisis de la pila de errores. -Creación de Excepciones.-Lanzamiento de Excepciones. | Construir aplicaciones con conexión a base de datos.Recuperación de la información en la base de datos desde la aplicación con programación orientada a objetos.Determinar los tipos de errores.Determinar los tipos de excepciones.Determinar el manejo de excepciones en la ejecución de programa. |

|  |
| --- |
| **Proceso Enseñanza-Aprendizaje** |
| Métodos y técnicas de enseñanza | Medios y materiales didácticos | Espacio Formativo |
| **Aula** |  |
| Aprendizaje basado en proyectosEquipos cooperativosResolver situaciones problemáticas | ProyectorPizarrónBibliografíaBuscadores académicosLecciones en un LMSAcceso a internetSoftware de desarrollo | **Laboratorio / Taller** | X |
|  |  | **Empresa** |  |

|  |
| --- |
| **Proceso de Evaluación** |
| Resultado de Aprendizaje | Evidencia de Aprendizaje | Instrumentos de evaluación |
| Los estudiantes analizan aplicaciones con interfaz gráfica de usuario y con conexión a base de datos para resolver problemas del mundo real. | A partir de un portafolio de evidencias de prácticas realizar aplicaciones con interfaz gráfica de usuario y con conexión a base de datos resolviendo algún problema del mundo real. | Estudios de casosProyectos grupales y/o individuales |

|  |
| --- |
| **Perfil idóneo del docente** |
| **Formación académica** | **Formación Pedagógica** | **Experiencia Profesional** |
| Licenciatura en Ingeniería en Tecnologías de la Información, Licenciatura en Ingeniería en Sistemas computacionales, Ingeniería Informática, Ingeniería de Software o carrera afín. | Educación Basada en Competencias. Dominio de técnicas de enseñanza-aprendizaje.Técnicas de manejo de grupo.Capacidad para gestionar equipos de trabajo colaborativo. | Experiencia docente en asignaturas de programación orientada a objetos. Deseable experiencia como desarrollador de software o ingeniero de software. Participación en proyectos en la industria. Certificaciones o cursos relacionados con programación. |

|  |
| --- |
| **Referencias bibliográficas** |
| Autor | Año | Título del documento | Lugar de publicación | Editorial | ISBN |
| Boucheny, Vicent | 2021 | Aprender la programación orientada a objetos con el lenguaje Python (con ejercicios prácticos y corregidos) | Barcelona | ENI | 9782409031564 |
| Francisco Blasco | 2019 | Programación Orientada a Objetos en JAVA | Madrid | Ra-Ma | 9788499648057 |
| Fco. Javier Ceballos Sierra | 2018 | Programación orientada a objetos con C++ | España | Ra-Ma | 9788499647548 |
| Paul J. Deitel, Harvey M. Deitel | 2012 | Cómo programar en Java | México | Pearson Educación | 9786073211505 |

|  |
| --- |
| **Referencias digitales** |
| Autor | Fecha de recuperación | Título del documento | Vínculo |
| Calzadilla, J. C. F | Abril, 2024 | Sistemas de tareas docentes para la enseñanza y aprendizaje de la Programación Orientada a Objetos. | <https://rus.ucf.edu.cu/index.php/rus/article/view/3479> |
| Leonardo Bermón Angarita | Abril, 2024 | Ejercicios de programación orientada a objetos con Java y uml | <https://fadmon.unal.edu.co/fileadmin/user_upload/investigacion/centro_editorial/libros/ejercicios%20de%20programacion.pdf> |
| Manel Guerrero Zapata | Abril, 2024 | Colección de problemas de POO | <https://personals.ac.upc.edu/guerrero/poo/Problemas_POO.pdf> |
| Carmen Cerón Garnica | Abril, 2024 | POO en Java | <https://ecosistema.buap.mx/forms/files/dspace-23/index.html> |