

## PROGRAMA ANALÍTICO 2023

|  |                            |
|--|----------------------------|
| <b>CARRERA:</b> Tecnicatura Universitaria en Desarrollo de Software  | <b>SEDE:</b> Este          |
| <b>Espacio Curricular:</b> Sistemas y Paradigmas de Programación   |                            |
| <b>Campo de Formación:</b> Propio de la Familia Profesional  | <b>Ciclo Lectivo:</b> 2023 |
| <b>Área de Conocimiento:</b> Fundamentos científico-tecnológicos   | <b>Año de cursado:</b> 1°  |
| <b>Profesor/a:</b> Prades, Mauricio  |                            |
| <p><b>Evaluación:</b></p> <p><b>Características del espacio curricular:</b> Semestral</p> <p><b>Criterios:</b></p> <p>Se tomarán:</p> <ul style="list-style-type: none"> <li>○ 2 evaluaciones de <b>proceso en cada módulo del espacio curricular</b>. Cada trabajo práctico tendrá su recuperatorio.</li> <li>○ Una evaluación Global Integradora (EGI): examen que contendrá todos los contenidos básicos para cumplir con los indicadores de logro de los módulos.</li> <li>○ Este espacio curricular <b>SI</b> admite la condición de estudiante libre.</li> </ul> <p><i>La calificación final de proceso se obtiene promediando la totalidad de las evaluaciones de proceso.</i></p> <p><i>Para poder acceder la evaluación global integradora, todas las evaluaciones de proceso deben estar aprobadas.</i></p> <p><b>La EGI NO cuenta con instancia de recuperación.</b></p> <p><b><u>Instrumentos:</u></b></p> <ul style="list-style-type: none"> <li>○ Realización de trabajos prácticos en aula.</li> <li>○ Evaluación integradora escrita y práctica.</li> <li>○ Exposiciones grupales</li> </ul> <p><b><u>Porcentaje de Asistencia mínimo:</u> 50%</b></p> |                            |

**Módulo 1: Conocimiento y diferenciación de las distintas herramientas que se ocupan en la formas de programación**

**Formato:** Aula - Taller

**Carga horaria:** 51 horas

**Indicadores de Logro:**

- Identifica las principales características de implementación de un lenguaje de programación.
- Interpreta especificaciones sintácticas de un lenguaje.
- Reconoce el funcionamiento distinguiendo cada una de sus partes.
- Interpreta a la programación AOP, POO y Programación Funcional.

| Conocimientos   | Habilidades  | Actitudes   |
|---|--|---|
| <p>Lenguajes de Programación.<br/>Lenguajes compilados e interpretados, declarativos e imperativos</p> <p>Distintos Paradigmas de programación:<br/>Orientada a Objetos (POO), Programación Orientada a Aspectos (AOP). Estructurado y Funcional</p> <p>Sintaxis de los lenguajes. Expresiones Regulares.<br/>Semántica: significado de una expresión. Equivalencia de expresiones. Tipos de semántica. Ejemplos</p> <p>Elementos Constitutivos de un Lenguaje.<br/>Variables: sus atributos. Alcance y vida. Tipos de variables, visibilidad y alcance.<br/>Rutinas: Atributos de las rutinas. Tipos de paso de parámetros. Alcance. Formas de agruparse. Genericidad de las rutinas. Paralelismo. Threads. Estructuras de control de la concurrencia. Estructuras de control. Excepciones</p> <p>Orientación a Objetos. Clases, objetos y tipos de herencia. Diferencia entre sistema de tipos y sistema de clases. Encapsulamiento de los datos. Polimorfismo y genericidad de métodos. Introducción a patrones de Diseño. Estudio de un lenguaje orientado a objetos y ejercitación</p> <p>Lenguajes Funcionales. Introducción.</p> | <p>Identificar las principales características de implementación de un lenguaje de programación</p> <p>Modelizar artefactos de software a partir de especificaciones.</p> <p>Diseñar algoritmos e implementarlos en pseudocódigo</p> <p>Documentar sus actividades y los resultados obtenidos aportando elementos para asegurar la calidad de los proyectos</p> <p>Interpretar especificaciones sintácticas de un lenguaje.</p> <p>Interpretar la programación AOP, POO y Programación Funcional.<br/>Producir software a través de diagramas de clases.</p> | <p>Participa como miembro en equipos de trabajo para el desarrollo y/o mantenimiento de software y sistemas.</p> <p>Tenacidad y perseverancia en la búsqueda de soluciones a los problemas.</p> <p>Colabora en el trabajo con sus pares para lograr las metas de aprendizaje.</p> |

**Estrategias metodológicas:**

- Exposición dialogada.
- Resolución de ejercicios y problemas.
- Modalidad presencial y virtual.

**Recursos y materiales:**

- **Recursos:** pizarra, marcadores, Proyector, sala de reuniones de Google Meet, Internet, etc.
- **Materiales de estudio:** apuntes de la cátedra, guías de trabajos prácticos, bibliografía, documentación obtenida en Internet.

**Bibliografía:**

“Lenguajes de Programación – Principios y Práctica”, K. Louden. Editor: Thomson (2004)

“Programming Language Design Concepts”, D.Watt. Editor: Wiley (2004)

“Concepts, Techniques and Models of Computer Programming”, P.Van Roy, S. Haridi. Editor: MIT Press (2004)