

## PROGRAMA ANALÍTICO

### 1. DATOS INFORMATIVOS

<b>DEPARTAMENTO:</b> CIENCIAS DE LA COMPUTACION		<b>ÁREA DE CONOCIMIENTO:</b> PROGRAMACION	
<b>NOMBRE DE LA ASIGNATURA:</b> PROG ORIENTADA A OBJETOS		<b>PERIODO ACADÉMICO:</b> PREGRADO S-II OCT18-FEB19	
<b>CÓDIGO:</b> SWU03		<b>No. CREDITOS:</b>	<b>NIVEL:</b> PREGRADO
<b>FECHA ELABORACIÓN:</b>  31/10/2018	<b>EJE DE FORMACIÓN</b>	<b>HORAS / SEMANA</b>	
	BÁSICA	<b>TEÓRICAS:</b>	<b>PRÁCTICAS/LABORATORIO</b>
<b>DESCRIPCIÓN DE LA ASIGNATURA:</b> La materia Programación Orientada a Objetos, es una asignatura del eje de formación profesional, que se caracteriza por contribuir a la formación de los elementos de competencia y fortalecer las unidades de competencia en análisis, diseño, y construcción de aplicaciones de software, basado en el paradigma orientado a objetos, sus fundamentos y principios, como el encapsulamiento, la abstracción, la herencia, el polimorfismo apoyados, por el lenguaje de programación Java. Esta asignatura se enfoca principalmente en la resolución de problemas complejos del mundo real, y en producir aplicaciones de calidad, empleando principios y prácticas de la Ingeniería de Software, tales como pruebas de unidad, patrones de diseño y los "SOLID Principales". Se fortalece también con el uso de interfaces gráficas de usuario, y conexión a bases de datos que permiten la adecuada interacción entre el usuario y el computador.			
<b>CONTRIBUCIÓN DE LA ASIGNATURA A LA FORMACIÓN PROFESIONAL:</b> La asignatura contribuye al resultado de aprendizaje del nivel y es parte sustancial de la formación profesional, los componentes son la solución a problemas orientados a la integración de diferentes aplicaciones e infraestructura tecnológica existente en las organizaciones, bajo el sustento de la programación de computadores.			
<b>RESULTADO DE APRENDIZAJE DE LA CARRERA (UNIDAD DE COMPETENCIA):</b> Aplica el paradigma de programación orientado a objetos para implementar algoritmos en lenguajes de programación que solucionan problemas básicos en diferentes dominios			
<b>OBJETIVO DE LA ASIGNATURA:</b> Formar profesionales en Ingeniería de Software capaces de desarrollar sistemas informáticos mediante el uso de metodologías, herramientas y estándares, demostrando creatividad, eficiencia, eficacia y responsabilidad profesional; con el propósito de optimizar procesos, generar fuentes de empleo y contribuir en la mejora de la economía y competitividad de los sectores productivos del País.			
<b>RESULTADO DE APRENDIZAJE DE LA ASIGNATURA: (ELEMENTO DE COMPETENCIA):</b> Conceptuales: Conoce el paradigma basado en objetos y sus diferentes componentes. Procedimentales: Resuelve aplicaciones computacionales enfocadas al paradigma basado en objetos. Programación de aplicaciones utilizando el Paradigma Orientado a Objetos Actitudinales: Lidera el Trabajo en Equipo con pro-actividad, para el desarrollo de aplicaciones computacionales.			

### 2. SISTEMA DE CONTENIDOS Y RESULTADOS DEL APRENDIZAJE

UNIDADES DE CONTENIDOS	
<b>Unidad 1</b>  CONCEPTOS BÁSICOS DEL PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS Y PRINCIPIOS DE DISEÑO	<b>Resultados de Aprendizaje de la Unidad 1</b>  Aplica, utiliza técnicas de programación orientada a objetos (POO), emplea UML para el modelamiento de Clases (Diagramas de clases) y de requerimientos (diagramas de casos de uso), utilizando una herramienta de modelado y un lenguaje POO. Realiza pruebas de unidad y depuraciones de la aplicación.
<b>Sistemas de control de versionamiento (VCS)</b> 1.1.1 Software VCS: Git, GitHub.	
<b>Paradigmas de programación</b> 1.2.1 Transición de paradigma	
<b>Entorno de Desarrollo</b> Características e Instalación Administración y configuración del área de trabajo. Líneas de Comando	

# PROGRAMA ANALÍTICO

## UNIDADES DE CONTENIDOS

### **Revisión de conceptos generales de la POO.**

Principios Generales de la Programación Orientada a Objetos.  
Definición de clases, objetos, atributos y métodos

### **Modelamiento de clases y Objetos**

Modelamiento de clases y Objetos  
UML Diagramas de Casos de Uso, Clases.  
Identificar clases de un sistema. Uso correcto de identificadores  
Modificadores de Acceso  
Implementación de clases

### **Estructura general de un programa**

Creación de un programa básico O.O  
Tipos de datos, primitivos y referencia dos.

### **Constructores**

Tipos de constructores  
Instanciación

### **Métodos getters, setters.**

Definición  
Aplicación

### **Excepciones**

Definición  
Excepciones y errores  
Clases de excepción  
Tipos de excepciones  
Propagación de excepciones  
Lanzamiento de excepciones  
Métodos para el control de excepciones  
Excepciones personalizadas.

### **Lectura escritura de datos por consola**

Entrada de datos  
Salida de datos

### **Arreglos estáticos y dinámicos.**

arreglos de datos primitivos  
Arreglos de objetos  
colecciones

### **Relaciones entre clases**

asociación, Agregación / Composición: modelado e implementación  
Dependencia: Modelado

### **Generalización/Especialización**

Herencia: Definición, ventajas, nomenclatura, reglas y modelado.  
Implementación de herencia.

### **Código limpio**

Estándares de implementación, buenas prácticas de programación.  
Atributos de calidad de código

### **Gestión de Defectos (testing).**

verificación y Validación  
Pruebas Vs Depuración.  
Pruebas de unidad

# PROGRAMA ANALÍTICO

UNIDADES DE CONTENIDOS	
<b>Unidad 2</b>  FUNDAMENTOS DE LA POO y PRINCIPIOS DE LA PROGRAMACIÓN VISUAL	<b>Resultados de Aprendizaje de la Unidad 2</b>  Diseña aplicaciones enfocadas a la ingeniería con técnicas de programación orientada a objetos con el uso de interfaz gráfica amigable para el usuario y persistencia de datos.
<p><b>Encapsulamiento</b></p> <ul style="list-style-type: none"> <li>definición</li> <li>clases internas</li> <li>paquetes</li> </ul> <p><b>polimorfismo</b></p> <ul style="list-style-type: none"> <li>Definición y ventajas</li> <li>sobrecargas de metodos</li> <li>sobre escritura de metodos</li> <li>asignación de objetos a variables de su superclase.</li> </ul> <p><b>Abstracción</b></p> <ul style="list-style-type: none"> <li>definicion</li> <li>modelado</li> <li>Clases abstractas</li> <li>Metodos abstractos</li> </ul> <p><b>Interfaces de programación</b></p> <ul style="list-style-type: none"> <li>definicion</li> <li>modelado</li> <li>Declaración e implementación</li> <li>Interfaces y polimorfismo</li> </ul> <p><b>Persistencia de datos</b></p> <ul style="list-style-type: none"> <li>Manipulación de archivos</li> <li>Lectura y escritura de datos primitivos</li> <li>Lectura y escritura de objetos: Serialización.</li> </ul> <p><b>Componentes y objetos gráficos</b></p> <ul style="list-style-type: none"> <li>Widgets (componentes gráficos)</li> <li>formularios</li> <li>Menús, tablas</li> <li>gestión de eventos</li> <li>Integración de componentes gráficos y clases</li> </ul>	

<b>Unidad 3</b>  PATRONES DE DISEÑO Y CONEXIÓN A BASES DE DATOS NO SQL	<b>Resultados de Aprendizaje de la Unidad 3</b>  Diseña e Implementa programas de computación con calidad utilizando la POO, interfaces gráficas, patrones de diseño y acceso a bases de datos. Aplica principios de POO (SOLID), patrones de diseño y fundamentos básicos de Ingeniería de Software.
<p><b>SOLID Principles</b></p> <ul style="list-style-type: none"> <li>Single Responsibility</li> <li>Open/Closed</li> <li>Liskov Substitution</li> <li>Interface Segregation</li> <li>Dependency Inversion</li> <li>Código entendible, flexible y mantenible</li> </ul> <p><b>modularización</b></p> <ul style="list-style-type: none"> <li>Localización de decisiones de diseño</li> <li>Alta cohesión</li> <li>bajo acoplamiento</li> </ul> <p><b>Introducción a Patrones de Diseño</b></p>	

# PROGRAMA ANALÍTICO

## UNIDADES DE CONTENIDOS

Conceptos Generales

Importancia de los patrones de diseño

tipos de patrones

### Patrones de creación

Singleton

Abstract Factory

### Patrones de estructura

Composite

template

### Patrones de comportamiento

Observer

Strategy

### Modelo Vista Controlador

arquitectura

Implementación

### Bases de Datos no SQL

Acceso a base de datos

Drivers y Conexión

Operaciones CRUD

## 3. PROYECCIÓN METODOLÓGICA Y ORGANIZATIVA PARA EL DESARROLLO DE LA ASIGNATURA

### (PROYECCIÓN DE LOS MÉTODOS DE ENSEÑANZA - APRENDIZAJE QUE SE UTILIZARÁN)

- 1 • A través de preguntas y participación de los estudiantes el docente recuerda los requisitos de aprendizaje previos que permite al docente conocer cuál es la línea de base a partir del cual incorporará nuevos elementos de competencia, en caso de encontrar deficiencias enviará tareas para atender los problemas individuales.
- 2 • Se diagnosticará conocimientos y habilidades adquiridas al iniciar el periodo académico.
- 3 • Con la ayuda del diagnóstico se indagará lo que conoce el estudiante, como lo relaciona, que puede hacer con la ayuda de otros, qué puede hacer solo, qué ha logrado y qué le falta para alcanzar su aprendizaje significativo.
- 4 • Plantear interrogantes a los estudiantes para que den sus criterios y puedan asimilar la situación problemática.
- 5 • Se iniciará con explicaciones orientadoras del contenido de estudio, donde el docente plantea los aspectos más significativos, los conceptos, leyes y principios y métodos esenciales; y propone la secuencia de trabajo en cada unidad de estudio.
- 6 • Se buscará que el aprendizaje se base en el análisis y solución de problemas; usando información en forma significativa; favoreciendo la retención; la comprensión; el uso o aplicación de la información, los conceptos, las ideas, los principios y las habilidades en la resolución de problemas de la programación orientada a objetos.
- 7 • Se buscará la resolución de casos para favorecer la realización de procesos de pensamiento complejo, tales como: análisis, razonamientos, argumentaciones, revisiones y profundización de diversos temas.
- 7 • Se realizan prácticas de laboratorio para desarrollar las habilidades proyectadas en función de las competencias y el desarrollo de proyectos aplicativos, que fortalezcan la consecución de destrezas de desarrollo de software.
- 8 • Se realizan ejercicios orientados a la carrera y otros propios del campo de estudio.
- 8 La evaluación cumplirá con las tres fases: diagnóstica, formativa y sumativa, valorando el desarrollo del estudiante en cada tarea y en especial en las evidencias del aprendizaje de cada unidad;

### PROYECCIÓN DEL EMPLEO DE LA TIC EN LOS PROCESOS DE APRENDIZAJE

- 1 • Para optimizar el proceso de enseñanza-aprendizaje, se utilizará el laboratorio con el apoyo de software con pertinencia, actualidad, eficiencia, eficacia, basados en los procesos de aprendizaje en sus diferentes fases: fase de familiarización del conocimiento, fase de reproducción del conocimiento y principalmente proyectando a un aprendizaje productivo y creativo donde se expresen, verifiquen y evidencien los conocimientos adquiridos.
- 2 • Las TIC, tecnologías de la información y la comunicación, se las emplearán para realizar las simulaciones de los temas tratados en el aula y presentaciones.
- 3 • Se utilizarán un lenguaje de programación Orientado a Objetos.
- 4 Además, los estudiantes deben tener las competencias para resolver aplicativamente requerimientos de modelamiento de clases, multiprocesamiento, manejo de interfaces gráficas, almacenamiento eficiente de datos y modelos básicos de persistencia.

# PROGRAMA ANALÍTICO

## 4. TÉCNICAS Y PONDERACIÓN DE LA EVALUACIÓN

- En este espacio se expresarán las técnicas utilizadas en la evaluación del proceso de enseñanza aprendizaje o evaluación formativa y sumativa.
- Las técnicas que se recomienda usar son: Resolución de ejercicios, Investigación Bibliográfica, Lecciones oral/escrita, Pruebas orales/escrita, Laboratorios, Talleres, Solución de problemas, Prácticas, Exposición, Trabajo colaborativo, Examen parcial, Otras formas de evaluación.
- Recordar que mientras más técnicas utilicen, la evaluación será más objetiva y el desempeño del estudiante se reflejará en su rendimiento (4 o 5 técnicas).
- Para evaluar se deberá aplicar la rúbrica en cada una de las técnicas de evaluación empleadas. Se debe expresar en puntaje de la nota final sobre 20 puntos. No debe existir una diferencia mayor a dos puntos entre cada técnica de evaluación empleada.
- En la modalidad presencial existen tres parciales en la modalidad a distancia existen dos parciales, toda la planificación de periodo académico se la realiza en función del número de parciales de cada modalidad.
- La ponderación a utilizarse en la evaluación del aprendizaje del estudiante será la misma en las tres parciales.
- Para la aprobación de una asignatura se debe tener una nota final promedio de 14/20, en los tres o dos

## 5. BIBLIOGRAFÍA BÁSICA/ TEXTO GUÍA DE LA ASIGNATURA

Titulo	Autor	Edición	Año	Idioma	Editorial
Manual imprescindible Java 8 : método experimentado para estudiantes de programación en Java con más de 200 ejemplos prácticos	Moldes Teo, F. Javier	1	2014	spa	Anaya Multimedia
Estructuras de datos básicas: programación orientada a objetos con Java	Guardati Buemo, Silvia		2015	spa	Alfaomega
Aprender a programar con Java : un enfoque práctico partiendo de cero	Jiménez Marín, Alfonso	2	2016	spa	Paraninfo

## 6. FIRMAS DE LEGALIZACIÓN

---

FRANKLIN JAVIER MONTALUISA YUGLA  
COORDINADOR DE AREA DE CONOCIMIENTO

---

DIRECTOR DE CARRERA

---

FABIÁN ARMANDO ÁLVAREZ SALAZAR  
DIRECTOR DE DEPARTAMENTO