PROGRAMA ANALÍTICO

1. DATOS INFORMATIVOS

DEPARTAMENTO: CIENCIAS DE LA COMPUTACION		**ÁREA DE CONOCIMIENTO:** PROGRAMACION	
NOMBRE DE LA ASIGNATURA: PROG. ORIENTADA A OBJETOS		**PERIODO ACADÉMICO:** PREGRADO S-I MAY21 - SEP21	
CÓDIGO: A0J08		**No. CREDITOS:**	**NIVEL:** PREGRADO
FECHA ELABORACIÓN: 30/11/2020	**EJE DE FORMACIÓN** BÁSICA	**HORAS / SEMANA** **TEÓRICAS:**	**PRÁCTICAS/LABORATORIO**

DESCRIPCIÓN DE LA ASIGNATURA:

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.

CONTRIBUCIÓN DE LA ASIGNATURA A LA FORMACIÓN PROFESIONAL:

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.

RESULTADO DE APRENDIZAJE DE LA CARRERA (UNIDAD DE COMPETENCIA):

Aplica el paradigma de programación orientado a objetos para implementar algoritmos en lenguajes de programación que solucionan problemas básicos en diferentes dominios

OBJETIVO DE LA ASIGNATURA:

Brindar lo conocimientos esenciales al estudiante para que aprenda a analizar, diseñar, implementar y probar software usando el paradigma de Orientación a Objetos, y de esta manera sea capaz de emplear métodos, técnicas y herramientas ingenieriles para la construcción de aplicaciones robustas y mantenibles mediante el uso de este paradigma.

RESULTADO DE APRENDIZAJE DE LA ASIGNATURA: (ELEMENTO DE COMPETENCIA):

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	
Unidad 1 CONCEPTOS BÁSICOS DEL PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS Y PRINCIPIOS DE DISEÑO.	**Resultados de Aprendizaje de la Unidad 1** 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.
Sistemas de control de versionamiento (VCS) Software VCS: Git, GitHub **Paradigmas de programación** Transición de paradigma **Entorno de Desarrollo** Características e Instalación.	

PROGRAMA ANALÍTICO

UNIDADES DE CONTENIDOS

Administración y configuración del área de trabajo.

Líneas de Comando

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

UML: Diagramas de Casos de Uso

UML: Diagramas de Clases

Identificación de clases de un sistema, uso correcto de identificadores.

Modificadores de Acceso

Implementación de clases

Código limpio

Estándares de implementación, buenas prácticas de programación.

Atributos de calidad de código

Estructura general de un programa

Creación de un programa básico O.O

Tipos de datos, primitivos y referenciados.

Lectura escritura de datos por consola

Entrada de datos

Salida de datos

Excepciones

Definición

Excepciones y errores

Clases de excepción

Tipos de excepciones.

Excepciones personalizadas.

Persistencia de datos

Manipulación de archivos

Lectura y escritura de datos primitivos

Lectura y escritura de objetos: Serialización

Formatos de datos: csv, json, otros

Encapsulamiento

Definición

Clases

Paquetes

Librerías/Bibliotecas, métodos static

Constructores

Tipos de constructores

Instanciación

Métodos getters, setters.

Definición e Implementación

Arreglos y Colecciones.

Arreglos de datos primitivos

Arreglos de objetos.

Colecciones/ArrayList

Relaciones entre clases

Asociación, Agregación / Composición: modelado e implementación

Dependencia: Modelado

Generalización/Especialización

PROGRAMA ANALÍTICO

UNIDADES DE CONTENIDOS

Herencia: Definición, ventajas, nomenclatura, reglas y modelado.

Implementación.

Revisiones de Código

Revisiones de Código

Unidad 2	**Resultados de Aprendizaje de la Unidad 2**
PRINCIPIOS AVANZADOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS E INTEGRACIÓN DE OBJETOS GRAFICOS Y CONEXIÓN A BASES DE DATOS NO SQL	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.

Gestión de Defectos (testing).

Verificación y Validación

Pruebas Vs Depuración

Pruebas de unidad

Polimorfismo

Definición y ventajas

Sobrecarga de métodos

Sobre escritura de métodos

Asignación de objetos a variables de su superclase

Abstracción

Definición

Modelado

Clases abstractas

Métodos abstractos

Interfaces de programación

Definición

Modelado

Declaración e implementación

Interfaces y polimorfismo.

Modelo Vista Controlador

Arquitectura

Implementación

Bases de Datos no SQL

Acceso a base de datos

Drivers y Conexión

Operaciones CRUD

Componentes y objetos gráficos

Widgets (componentes gráficos)

Formularios

Menús, tablas

Gestión de eventos

Integración de componentes gráficos y clases

Unidad 3	**Resultados de Aprendizaje de la Unidad 3**
TECNICA PARA RESOLVER PROBLEMAS EN EL DESARROLLO DE SOFTWARE	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.

SOLID Principles

Single Responsibility

Open/Closed

Liskov Substitution

Interface Segregation

Dependency Inversion

PROGRAMA ANALÍTICO

UNIDADES DE CONTENIDOS

Código entendible, flexible y mantenible

Modularidad

Localización de decisiones de diseño

Alta cohesión

Bajo acoplamiento

Introducción a Patrones de Diseño

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

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

(PROYECCIÓN DE LOS MÉTODOS DE ENZEÑANZA - APRENDIZAJE QUE SE UTILIZARÁN)

1 Clase Magistral

2 Resolución de Problemas

3 Diseño de proyectos, modelos y prototipos

4 Prácticas de Laboratorío

PROYECCIÓN DEL EMPLEO DE LA TIC EN LOS PROCESOS DE APRENDIZAJE

1 Herramientas Colaborativas (Google, drive, onedrives, otros)

2 Video Conferencia

3 Aula Virtual

4 Material Multimedia

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
Introducción a la programación orientada a objetos	Deitel, Paul J	-	2010	spa	México : Pearson
Programación orientada a objetos y	Pérez, María	-	2014	spa	Estados Unidos de

PROGRAMA ANALÍTICO

Titulo	Autor	Edición	Año	Idioma	Editorial
programación estructurada	Pérez, María	-	2014	spa	América : [s.n.]

6. FIRMAS DE LEGALIZACIÓN

FRANKLIN JAVIER MONTALUISA YUGLA
COORDINADOR DE AREA DE CONOCIMIENTO

DIRECTOR DE CARRERA

FABIÁN ARMANDO ÁLVAREZ SALAZAR
DIRECTOR DE DEPARTAMENTO