**ACTIVIDAD DE APRENDIZAJE**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sigla Asignatura** | | **POO2201** | **Nombre Asignatura** | | | **Programación Orientada a Objetos I** | | | |
| Créditos | 10 | Hrs. Semestrales Totales | | 90 | Requisitos | | POO1501 | Fecha Actualización | 23-ABRIL-13 |

|  |  |  |  |
| --- | --- | --- | --- |
| Escuela o Programa Transversal | ESCUELA DE INFORMÁTICA Y TELECOMUNICACIONES | Currículum | 1446703 |
| Carrera/s | **INGENIERÍA EN INFORMÁTICA**  **INGENIERÍA EN GESTIÓN DE TECNOLOGÍAS DE INFORMACIÓN**  **ANALISTA PROGRAMADOR COMPUTACIONAL** | Unidad de Aprendizaje | **Programación de interfaces gráficas de usuarios** |

|  |  |  |
| --- | --- | --- |
| **APRENDIZAJE(S) ESPERADO(S)**  Construye una interfaz gráfica de usuario para que resuelva un problema.  Construye casos de prueba utilizando el framework JUnit para verificar el correcto funcionamiento.  Revisa la codificación de clases programadas en Java utilizando la herramienta de debugging del entorno de desarrollo para detectar errores. | | |
| **NOMBRE DE LA ACTIVIDAD**  Biblioteca | | |
| **Ambiente de aprendizaje**  √ Presencial  □ No Presencial  □ Mixta | **Duración de la actividad de aprendizaje (horas pedagógicas presenciales y no presenciales):**  5 horas  Desarrollo de la actividad 3 horas  Heteroevaluación 1 hora  Ejecución del docente y conclusiones 1 hora | |
| **Forma de trabajo:**  √ Individual  □ Grupal   * Tamaño del grupo:   □ 2 □ 3-5 □ 6-8 | **Forma de retroalimentación:**  **¿Quién evalúa?**  □ Estudiante  √ Pares  □ Docente  **¿Cómo se evalúa?**  √ Rúbrica  □ Escala de valoración  □ Lista de cotejo  Otra: | |
| **Infraestructura (lugar):**  □ Sala de clases  √ Laboratorio con Netbeans  □ Taller  □ Terreno  □ Otros | | **Insumos y equipamiento para la actividad de aprendizaje:**  PC |
| **Recursos de información:**  √ Textos  □ Recursos audiovisuales  □ Recursos web  √ Programas computacionales: NetBeans | | **Recursos didácticos:**  □ Textos  √ Recursos audiovisuales: Video del programa  □ Recursos web  √ Programas computacionales: Pseudocódigo fuente |
| **DESCRIPCIÓN DE LA ACTIVIDAD: Caso e Instrucciones**  **CASO**  En una biblioteca universitaria se está construyendo un sistema de control de los distintos documentos, como libros y revistas. La forma de acceso es: consultas en sala para las revistas y libros y préstamos temporales para los libros.  Los tipos de documentos que se contemplan son:  Los libros clásicos, en papel. Los datos que interesa conocer son: código, título, autor o autores, editorial, año de publicación y si se encuentra en préstamo (true) o no (false). De estos libros algunos se pueden prestar y otros sólo se pueden consultar en la sala.  Las revistas en papel que tienen las siguientes características: código, título, autor o autores, editorial, año de publicación, volumen, número y mes de salida. Sólo se pueden consultar en sala.  Diseñe las clases y relaciones que representen una solución para este problema. El diagrama de clases es:    Las clases deben tener constructor con parámetros, accesadores, mutadores y método que retorne todos sus atributos.  La interfaz gráfica debe partir de un menú que tenga las siguientes opciones:    El sistema debe cumplir las siguientes restricciones y operaciones:   * La opción salir del menú permite terminar la aplicación. * La opción Agregar permite ingresar a la ventana de ingreso de un documento. * La opción Eliminar permite ingresar a la ventana de eliminación de un documento por su código. * La opción Listar permite mostrar todos los documentos almacenados en una colección. * La ventana de ingreso se debe visualizar:     El comboBox Documento tiene las opciones: Seleccione, Libro y Revista.  La ventana parte con los campos volumen, número, mes y préstamo deshabilitados. El volumen,  número y mes se habilitan cuando se selecciona Revista y préstamo se habilita cuando se selecciona  Libro.   * El botón ingresar almacena los Libros y Revistas en una colección validando previamente que se haya seleccionado un Documento. * El botón limpiar limpia los campos de texto, los comboBox los deja en seleccione y el préstamo en Si. Además, deshabilita los campos volumen, número, mes y préstamo, dejando la ventana en su estado inicial. * El botón Salir cierra la ventana y deja activo el menú. * La ventana eliminar tiene un botón eliminar y permite borrar un Documento por su código:      * La ventana listar tiene un botón listar y permite mostrar todos los documentos almacenados en la   colección:     * Debe existir una clase que tenga como atributo la colección y los métodos que permitan: agregar   documentos a la colección validando previamente que el código no exista, listar documentos y  eliminar documento. | | |