|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **INSTRUMENTOS DE EVALUACIÓN** | | |  |
| Código: FO-FIT-129 | Versión: 001 | Página 1 de 1 |

|  |  |
| --- | --- |
| **Instrumento de Evaluación:** | Taller de POO |
| **Nombre de Estudiante:** |  |
| **Guía de Aprendizaje:** | Introducción a POO |
| **Fecha:** |  |
| **Jornada:** |  |
| **Nombre del Docente:** | James Mosquera Rentería |

**Descripción**: En este instrumento, los estudiantes deben desarrollar un proyecto práctico que aplique los conceptos de la POO en Python. A continuación, se describe un ejemplo de proyecto:

**Proyecto: Sistema de Gestión de Biblioteca en Python**

**Requisitos del Proyecto**:

* Cree una clase **Libro** que tenga atributos como título, autor, ISBN y disponibilidad.

class Libros:

    titulo = "titulo"

    autor = "autor"

    isbn = "isbn"

    disponible = "disponible"

biblioteca=Libros()

print(biblioteca.titulo)

print(biblioteca.autor)

print(biblioteca.isbn)

print(biblioteca.disponible)

* Implemente un método en la clase **Libro** que permita prestar un libro y actualizar su disponibilidad.

class Libros:

    def \_\_init\_\_(self, titulo, autor, isbn, disponibilidad):

        self.titulo = titulo

        self.autor = autor

        self.isbn = isbn

        self.disponibilidad = disponibilidad

    def prestar\_libro(self):

        print(f"Prestar libros... \nTitulo: {self.titulo} \nAutor: {self.autor} \nISBN: {self.isbn} \nDisponibilidad: {self.disponibilidad}")

        self.disponibilidad = "no disponible"

biblioteca=Libros("New York Times", "Carmenza", 99891223, "Disponible")

biblioteca.prestar\_libro()

* Cree una clase **Biblioteca** que contenga una lista de libros y permita agregar libros a la biblioteca.

class Biblioteca:

    def \_\_init\_\_(self):

        self.libros = []

    def agregar\_libro(self, libro):

        self.libros.append(libro)

    def \_\_str\_\_(self) -> str:

        return f"Libros: {self.libros}"

biblioteca1 = Biblioteca()

biblioteca1.agregar\_libro(input("Ingrese el nombre del libro: "))

print(biblioteca1)

biblioteca1.agregar\_libro(input("Ingrese el nombre del libro: "))

print(biblioteca1)

* Implemente un método en la clase **Biblioteca** que muestre la lista de libros disponibles en la biblioteca.

class Biblioteca:

    def \_\_init\_\_(self):

        self.libros = []

        self.disponible = "Disponible"

    def agregar\_libro(self, libro):

        self.libros.append(libro)

    def mostrar\_libros\_disponibles(self):

        """

        Muestra la lista de libros disponibles en la biblioteca.

        """

        for libro in self.libros:

            if libro[3] == True:

                print(libro)

biblioteca = Biblioteca()

biblioteca.agregar\_libro(["El Quijote", "Miguel de Cervantes", "978-84-376-0494-8", True])

biblioteca.agregar\_libro(["Cien años de soledad", "Gabriel García Márquez", "978-84-339-0040-9", True])

biblioteca.agregar\_libro(["La Biblia", "Varios autores", "978-84-283-1531-2", False])

biblioteca.mostrar\_libros\_disponibles()

**Entrega**:

**Enlace de GitHub con el código del proyecto**

|  |  |  |  |
| --- | --- | --- | --- |
| **N.º** | **Aspecto para evaluar** | **Si** | **No** |
| 1 | Creación de Clases: El estudiante debe crear las clases Libro y Biblioteca en Python. |  |  |
| 2 | Definición de Atributos: Los atributos de la clase Libro deben ser definidos correctamente. |  |  |
| 3 | Implementación de Métodos: Los métodos para prestar libros, agregar libros y mostrar la lista de libros disponibles en la clase Biblioteca deben funcionar correctamente. |  |  |
| 4 | Aplicación de Conceptos de POO: aplicación de conceptos como encapsulamiento, herencia, y métodos de clases. |  |  |
|  | Funcionalidad: La aplicación debe cumplir con los requisitos del proyecto y ser funcional. |  |  |
|  | Organización y Comentarios: Se evaluará la estructura organizativa del código y la presencia de comentarios explicativos. |  |  |

**Firma del estudiante Firma del instructor**