

## Autoría:

**Recurso producido por:** Denilson Florentín de León Aguilar.

**Como parte del curso:** Introducción a la Programación y Computación 2 Sección P S2 2024.

---

## Descripción del Problema

Se requiere desarrollar un sistema que permita almacenar y modificar los datos e información de estudiantes y cursos proporcionados en archivos XML. La aplicación debe ofrecer funcionalidades para cargar, visualizar, buscar y actualizar tanto la información de los estudiantes como la de los cursos. Para abordar este problema, se han definido las siguientes clases:

---

## Clases y Atributos

### Clase `Student` (dao)

#### Atributos

- `dpi` : Documento Personal de Identificación del estudiante (entero)
- `name` : Nombre del estudiante (cadena de texto)
- `last_name` : Apellido del estudiante (cadena de texto)
- `email` : Dirección de correo electrónico del estudiante (cadena de texto)
- `gender` : Género del estudiante (cadena de texto)
- `inscrito` : Estado de inscripción del estudiante (booleano)

### Clase `Course` (dao)

#### Atributos

- `code` : Código identificador del curso (entero)
- `name` : Nombre del curso (cadena de texto)
- `credits_value` : Valor de créditos del curso (entero)

### Clase `EntityController` (controller)

#### Métodos Abstractos

- `entity_to_xml_element(self, entity)` : Método para convertir una entidad (estudiante o curso) a un elemento XML.
- `xml_element_to_entity(self, xml_element)` : Método para convertir un elemento XML a una entidad.
- `xml_element_list_to_string(self, xml_elements)` : Método para convertir una lista de elementos XML a una cadena de texto.
- `xml_element_to_string(self, xml_element)` : Método para convertir un elemento XML a una cadena de texto.

### Clase `StudentController` (controller)

#### Métodos

- `xml_element_list_to_string(self, xml_elements)` : Método para convertir una lista de elementos XML de estudiantes a una cadena de texto.
- `xml_element_to_string(self, xml_element)` : Método para convertir un elemento XML de estudiante a una cadena de texto.

- `get_student_by_dpi(self, xml_element, dpi)` : Método para obtener un estudiante por su DPI.
- Métodos de actualización para atributos específicos (nombre, apellido, correo, género, inscripción).

## Clase `CourseController` (`controller`)

### Métodos

- `xml_element_list_to_string(self, xml_elements)` : Método para convertir una lista de elementos XML de cursos a una cadena de texto.
- `xml_element_to_string(self, xml_element)` : Método para convertir un elemento XML de curso a una cadena de texto.
- `get_course_by_code(self, xml_element, code)` : Método para obtener un curso por su código.
- Métodos de actualización para atributos específicos (nombre del curso, valor de créditos).

## Clase `XMLHandler` (`xml_serv`)

Esta clase es una abstract, por lo que significa que sus hijas terminarán de implementar dichas funciones.

### Métodos

- `create_xml_node(self, name)` : Método para crear un nodo XML.
- `write_xml(self, doc, file_path)` : Método para escribir un documento XML en un archivo.
- `load_xml(self, file_path)` : Método para cargar un archivo XML y obtener el elemento raíz.
- `add_child_to_node(self, parent_doc, parent_node, child)` : Método para agregar un nodo hijo a un nodo padre.
- `add_text_to_node(self, parent_doc, parent_node, text)` : Método para agregar texto a un nodo.
- `create_and_append_node_with_text_to_node(self, parent_doc, parent_node, name, text)` : Método para crear y agregar un nodo con texto a un nodo padre.
- `print_xml_node(self, node, indent="")` : Método para imprimir un nodo XML de manera legible.
- `get_node_text(self, node)` : Método para obtener el texto de un nodo.
- `get_node_text_by_tag(self, node, tag_name)` : Método para obtener el texto de un nodo por su etiqueta.
- `modify_node_text(self, node, new_text)` : Método para modificar el texto de un nodo.
- `get_xml_node_by_tag(self, root_node, tag_name)` : Método para obtener un nodo por su etiqueta.
- `get_xml_node_list_by_tag(self, root_node, tag_name)` : Método para obtener una lista de nodos por su etiqueta.
- `update_xml_node_value_by_tag(self, root_node, tag_name, new_value)` : Método para actualizar el valor de un nodo por su etiqueta.

## Clase `XMLTree` (`xml_serv`)

### Métodos

- Hereda de `XMLHandler`
- Métodos equivalentes a `XMLDom` utilizando `ElementTree`.

### **Clase XMLDom (xml\_serv)**

- Hereda de XMLHandler
- Métodos equivalentes a ElementTree utilizando `XMLDom`.

### **Métodos Abstractos**

- Métodos abstractos que serán implementados por XMLDom y XMLTree.

### **Clase FileDialog (file)**

#### **Métodos**

- `select_file(self)`: Método para abrir un cuadro de diálogo y seleccionar un archivo.

### **Clase LoadXMLService (file)**

#### **Métodos**

- `__init__(self, xml_service, path_file=None)`: Constructor para inicializar el servicio de carga XML.
- `read_and_print_xml(self)`: Método para leer y imprimir un archivo XML.
- `read_xml(self)`: Método para leer un archivo XML y obtener el elemento raíz.
- `write_xml(self, xml_element)`: Método para escribir un elemento XML en un archivo.

---

Con estas clases y atributos, se busca proporcionar una estructura organizada y modular para abordar la gestión de estudiantes y cursos a