

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería

Programas de Ingeniero en Computación e Ingeniero en Software y Tecnologías Emergentes

INFORMACIÓN DE LA MATERIA

Nombre de la materia y clave: Lenguaje de Programación Python (36305).

Grupo y periodo: 532 (2022-2)

Profesor: Manuel Castañón Puga.

INFORMACIÓN DE LA ACTIVIDAD

Nombre de la actividad: Práctica de laboratorio 3.1.2 Clases y Objetos (Programación)

Lugar y fecha: A 6 de enero de 2023 en el Edificio 6E, Salón 204.

Carácter de la actividad: Individual

Participante(es): Emmanuel Alberto Gómez Cárdenas

REPORTE DE ACTIVIDADES

1. Utilice el repositorio en GitHub con el portafolio de prácticas de laboratorio que creó en la Meta 1.2.
2. Clone el repositorio en su computadora y agregue una carpeta de código para la Actividad de taller 1.3.2. Puede hacerlo utilizando una IDE (Visual Studio Code, PyCharm, etc.).

3. Escribir una biblioteca de código en python que implemente el modelo de clases y objetos desarrollado en la Actividad de taller 3.1.1. Puede ser un módulo o una colección de módulos según convenga.

implementacion.py

```
import datetime
from university import *

university = University('Universidad de Baja California')

faculty = Faculty('Facultad de Ciencias Químicas e Ingeniería')
university.add_faculty(faculty)

semester = Semester('2022-2')

course = Course('Lenguaje de Programación Python', semester)
faculty.add_course(course)

professor = Professor('Profesor Manuel Castañón Puga')
course.add_professor(professor)

student = Student('Emmanuel Gómez')
course.add_student(student)

exam = Exam(datetime.datetime(2022, 10, 15), course)
course.add_exam(exam)

grade = Grade(student, exam, 90.0)
exam.add_grade(grade)
student.add_grade(grade)

print('Universidad:', university.name)
print('Facultades:')
for faculty in university.faculties:
    print('-', faculty.name)
    print('  Cursos:')
    for course in faculty.courses:
        print('    -', course.name)
        print('      Profesores:')
        for professor in course.professors:
            print('        -', professor.name)
        print('      Estudiantes:')
        for student in course.students:
            print('        -', student.name)
        print('      Exámenes:')
        for exam in course.exams:
            print('        -', exam.date)
```

Biblioteca university.py

```
import datetime
class University:
    def __init__(self, name: str):
        self.name = name
        self.faculties = []
        self.students = []

    def add_faculty(self, faculty: 'Faculty'):
        self.faculties.append(faculty)

    def add_student(self, student: 'Student'):
        self.students.append(student)

class Faculty:
    def __init__(self, name: str):
        self.name = name
        self.courses = []
        self.professors = []

    def add_course(self, course: 'Course'):
        self.courses.append(course)

    def add_professor(self, professor: 'Professor'):
        self.professors.append(professor)

class Course:
    def __init__(self, name: str, semester: 'Semester'):
        self.name = name
        self.semester = semester
        self.professors = []
        self.students = []
        self.exams = []
        self.grades = []

    def add_professor(self, professor: 'Professor'):
        self.professors.append(professor)

    def add_student(self, student: 'Student'):
        self.students.append(student)

    def add_exam(self, exam: 'Exam'):
        self.exams.append(exam)

    def add_grade(self, grade: 'Grade'):
        self.grades.append(grade)

class Professor:
    def __init__(self, name: str):
        self.name = name
        self.courses = []

    def add_course(self, course: 'Course'):
        self.courses.append(course)

class Student:
    def __init__(self, name: str):
        self.name = name
        self.courses = []
        self.grades = []

    def add_course(self, course: 'Course'):
```

```

        self.courses.append(course)

    def add_grade(self, grade: 'Grade'):
        self.grades.append(grade)

class Semester:
    def __init__(self, name: str):
        self.name = name

class Exam:
    def __init__(self, date: datetime.datetime, course: 'Course'):
        self.date = date
        self.course = course
        self.students = []
        self.grades = []

    def add_student(self, student: 'Student'):
        self.students.append(student)

    def add_grade(self, grade: 'Grade'):
        self.grades.append(grade)

class Grade:
    def __init__(self, student: 'Student', exam: 'Exam', grade: float):
        self.student = student
        self.exam = exam
        self.grade = grade

```

4. Respalde (commit) y suba (push) su código en el repositorio de GitHub para hacer la entrega de la práctica.

URL del repositorio:

<https://github.com/LPP-FCQI-UABC/lpp-2022-2-532-portafolio-AlbGmx>

RESUMEN/REFLEXIÓN/CONCLUSIÓN

Esta practica nos permitio escribir una biblioteca de código en Python que implementa el modelo de clases y objetos UML. El uso de las tecnologias UML y python, proporcionan una gran cantidad de beneficios a largo plazo y es una excelente oportunidad para seguir aprendiendo sobre dichas teconlogias.

Doy fe de que toda la información dada es completa y correcta.

Nombre y firma del alumno.

Gómez, Emmanuel A.