"Programación en Python para Bachillerato: Fundamentos y Aplicaciones"

Antonio Boada Lastra

NNEE 4º

colegio CALASANZ SANTANDER

INDICE

1.- TÍTULO

"Programación en Python para Bachillerato: Fundamentos y Aplicaciones"

2.- RESUMEN

Este trabajo profundiza en el proyecto educativo de Bachiller en una materia actualmente desconocida para muchos y de gran importancia en la vida real. La competencia digital es hoy en día altamente requerida tanto en el ámbito académico como en el empresarial. La mayor parte de las carreras universitarias tocan esta competencia si bien son las de la rama de ciencias las que lo tratan en mayor profundidad.

El proyecto que se detalla y explica en las siguientes páginas, está destinado al alumnado de 2ºBachiller, pero puede proyectarse también al segundo ciclo de la Enseñanza Secundaria Obligatoria (ESO).

En líneas generales, en cada una de las sesiones en las que se divide el curso, están destinadas al aprendizaje básico del lenguaje por parte de los alumnos y a la realización de un proyecto práctico final en donde pongan en uso todo lo aprendido.

3.- INTRODUCCIÓN

La materia de Programación y Gestión de Datos viene a responder a una necesidad, que cuenta con un amplio consenso, de abordar estas disciplinas en los currículos educativos.

Las tecnologías de la información tienen un papel protagonista por su importancia en los modos de relación, en la innovación en todos los ámbitos sociales y económicos y en el acceso al conocimiento. El software y la gestión de los datos que se generan a través de su uso, constituyen una piedra angular del crecimiento económico y social. Formar personas competentes en estos ámbitos es fundamental para el desarrollo de profesiones, actuales y futuras, relacionadas con la ingeniería, la salud, el arte, la economía o el bienestar social.

Las tecnologías de la información en general y la Programación y Gestión de Datos en particular tienen en el Bachillerato el objetivo de preparar al alumnado para afrontar con éxito los principales retos y desafíos digitales de la sociedad, favoreciendo situaciones de aprendizaje relevantes que sean aplicables en

diversos ámbitos académicos, sociales o profesionales de sus vidas. Por ello, en su desarrollo curricular, junto a los descriptores operativos de la competencia clave de digitalización, se tratarán otros transversales relacionados con la de STEM, la de comunicación lingüística, la personal, social y de aprender a aprender, y la emprendedora. El objetivo es que los alumnos aumenten las competencias digitales adquiridas en etapas anteriores para integrarlas en cualquier disciplina.

4.- DESARROLLO

4.1.- Objetivos

Los objetivos que se plantean para llevar a cabo el proyecto son los siguientes:

- Introducir la programación con Python
- Desarrollar pensamiento algorítmico
- Aplicar estructuras de control y funciones
- Manipular datos y estructuras avanzadas
- Resolver problemas con programación

4.2.- Participantes

El presente proyecto, está pensado para alumnos de 2º de Bachillerato de la rama Científico-Tecnológica. Muchos de ellos y a pesar de estar inmersos en una sociedad completamente digitalizada, es la primera vez que se enfrentan a los retos que conlleva esta materia.

A pesar de estar pensado para estos alumnos, su ámbito de aplicación podría también ser el segundo ciclo de la Enseñanza Secundaria Obligatoria (ESO)

4.3.- Procedimiento

El proyecto, está pensado para una duración aproximada de 10 semanas y en él se tratan los aspectos fundamentales que hay que saber para comenzar a programar en lenguaje Python.

La elección de este lenguaje se debe principalmente a que se trata de un lenguaje potente, fácil de aprender y con infinitas posibilidades. Algunas de ellas las podemos concretar en:

a.) Sintaxis sencilla y fácil de aprender.

Estructura clara y cercana al lenguaje natural lo que facilita el aprendizaje. Se escribe menos código para hacer más cosas.

b.) Versatilidad y múltiples aplicaciones

Altamente usado en desarrollo web, ciencia de datos, inteligencia artificial, automatización, videojuegos….

Su compatibilidad es muy con múltiples sistemas operativos (Windows, macOS, Linux).

c.) Gran comunidad y soporte

En la actualidad, son millones de programadores los que usan Python, esto garantiza abundante documentación, foros, recursos y cursos gratuitos (MOOC, píldoras…).

d.) Amplia cantidad de librerías y frameworks

Python tiene bibliotecas para casi todo:

- **Ciencia de datos e IA**: NumPy, Pandas, TensorFlow
- **Desarrollo web**: Django, Flask
- **Automatización**: Selenium, PyAutoGUI
- **Ciberseguridad**: Scapy, PyCryptodome

e.) Lenguaje multiplataforma y de código abierto

Funciona en distintos sistemas sin modificar el código.

Es de código abierto, lo que permite usarlo sin costo alguno.

f.) Alta demanda en el mercado laboral

Python es uno de los lenguajes más solicitados en tecnología, ciencia de datos y desarrollo de software.

g.) Integración con otros lenguajes

Altamente compatible con C, C++, Java y JavaScript, facilitando proyectos complejos y la reutilización de código ya existente.

h.) Ideal para automatización y scripting

Permite crear scripts para tareas repetitivas, como enviar correos, procesar archivos y extraer datos de la web.

4.4.- Temario

El curso está dividido en un total de siete módulos que nombraremos y detallaremos a continuación:

Módulo 1: Introducción a Python (Semana 1-2)

-Instalación y configuración (IDLE, VS Code, Jupyter…)

-Sintaxis y primeros programas (print(), variables, tipos de datos…)

-Operadores y expresiones básicas

Módulo 2: Estructuras de Control (Semana 3-4)

-Condicionales (if, elif, else)

-Bucles (for, while)

-Ejercicios prácticos: Adivina el número, tablas de multiplicar

Módulo 3: Funciones y Módulos (Semana 5-6)

-Definir funciones (def) y parámetros.

-Retorno de valores.

-Importación de módulos (math, random).

-Proyecto: Calculadora de IMC

Módulo 4: Listas y Diccionarios (Semana 7)

-Listas: Métodos y manipulación.

-Diccionarios: Claves y valores.

-Ejercicio: Agenda de contactos

Módulo 5: Manejo de Archivos y Excepciones (Semana 8)

-Lectura y escritura de archivos (open, with).

-Manejo de errores (try-except).

-Proyecto: Registro de notas en un archivo

Módulo 6: Introducción a la Progr. Orientada a Objetos (Semana 9)

-Clases y objetos (class, __init__()).

-Métodos y atributos.

-Proyecto: Sistema de gestión de alumnos

Módulo 7: Proyecto Final (Semana 10)

Desarrollo de un mini-proyecto integrador a elección:

-Juego con pygame

-Chatbot simple

-Analizador de texto

-Aplicación con interfaz gráfica (tkinter)

4.5.- Metodología

La metodología que se sigue para el desarrollo del proyecto es:

a.) Clases prácticas: 70% código, 30% teoría.

b.) Ejercicios guiados.

c.) Trabajo en parejas y grupos

d.) Evaluación:

- 50% proyectos
- 30% participación
- 20% pruebas cortas

4.6.- Uso de metodologías activas

De manera general, para una clase de programación en Python en **bachillerato**, las **metodologías activas** son fundamentales entre otras cosas, para mantener el interés de los estudiantes, fomentar la participación y lograr un aprendizaje significativo. En concreto para el desarrollo de esta asignatura se han seguido las siguientes:

→Aprendizaje basado en proyectos (ABP)

- **Qué es**: Los estudiantes desarrollan un proyecto real o simulado con Python.
- **Ejemplos**:
  - Calculadora de IMC.
  - Registro de notas en un archivo.
  - Sistema de gestión de alumnos.
- **Ventajas:** Fomenta la autonomía, creatividad y aplicación práctica.

→Aprendizaje cooperativo

- **Qué es**: Los alumnos trabajan en parejas o pequeños grupos.
- **Aplicación**:
  - Resolver un problema juntos y repartir roles (uno escribe, otro revisa).
  - Programación en parejas (*pair programming*).
- **Ventajas:** Desarrolla habilidades sociales y resolución de problemas en equipo.

→Clase invertida (flipped classroom)

- **Qué es**: El contenido teórico se estudia en casa (videos, lecturas) propuestas a través del grupo de TEAMS de la asignatura. , y la clase se usa para resolver ejercicios prácticos.
- **Aplicación en Python**:
  - Video explicando if, for, def.
  - En clase: pequeños retos prácticos y juegos tipo quiz.
- **Ventajas**: Más tiempo para práctica guiada.

→Aprendizaje basado en problemas (ABP o PBL)

- **Qué es**: Partir de un problema realista para aprender nuevos conceptos.
- **Ejemplo**:
  - "Un comercio necesita automatizar el cálculo de descuentos. ¿Cómo lo harías con Python?"
- **Objetivo**: Los estudiantes investigan y aplican lo aprendido para resolverlo.

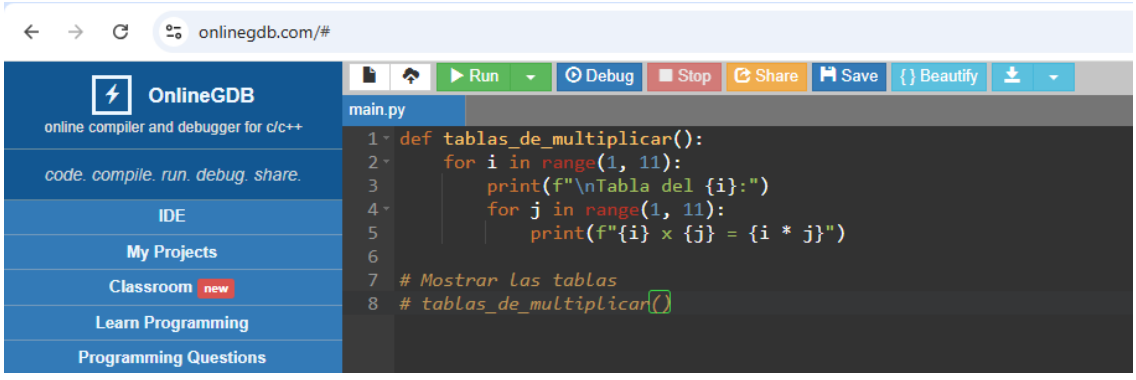
5.- ANEXOS

5.1.- ANEXO 1.-ALGUNOS EJEMPLOS DE CÓDIGO

A continuación se muestran algunas de las soluciones planteadas a los ejercicios de los diferentes módulos expuestos en el apartado 4.4 del presente documento:

1.- Adivinar un número aleatorio:

2.- Tablas demultiplicar usando funciones



5.2.- ANEXO 2.- COMPILADOR DE CÓDIGO

OnlineGDB es un entorno de desarrollo integrado (IDE) en línea gratuito que permite escribir, compilar, depurar y ejecutar programas en varios lenguajes de programación directamente desde un navegador web, sin necesidad de instalar ningún software en tu ordenador.

Características principales de OnlineGDB:

- **Compilación y ejecución online**: Soporta muchos lenguajes como C, C++, Python, Java, PHP, JavaScript, C#, y más.
- **Depurador integrado**: Puedes poner *breakpoints*, ejecutar paso a paso, ver variables, etc., como en un IDE de escritorio.
- **Interfaz sencilla**: Muy útil para estudiantes o quienes están empezando a programar.
- **Trabajo colaborativo**: Puedes compartir el enlace de tu código para que otros lo vean o editen.
- **Editor de código con resaltado de sintaxis**: Colorea tu código para facilitar su lectura.

Ejemplo de uso:

1. Vas a https://www.onlinegdb.com.
2. Eliges el lenguaje de programación (por ejemplo, Python).
3. Escribes tu código.
4. Haces clic en **"Run"** (Ejecutar) para ver los resultados.

¿Para qué sirve?

- Probar ideas rápidas.
- Aprender a programar sin instalar nada.
- Enseñar programación online.
- Resolver ejercicios desde cualquier lugar.

5.3.- ANEXO 3-IMÁGENES