



Programación Didáctica: Programación en Python para Administradores/as de Sistemas

1. Introducción

Esta programación didáctica deseñouse para o alumnado do ciclo de Administración de Sistemas Informáticos. O seu obxectivo estratéxico é dotar os futuros profesionais de competencias de programación directamente aplicables ao seu campo, permitíndolles automatizar tarefas, xestionar recursos de forma eficiente e resolver problemas complexos mediante o desenvolvemento de scripts e pequenas aplicacións.

A contorna de traballo inicial será unha máquina virtual en Windows, onde o alumnado instalará as ferramentas de desenvolvemento fundamentais. Progresivamente, a formación transitará cara a un sistema operativo GNU/Linux, o ecosistema principal onde se desenvolverán e aplicarán as tarefas de administración de sistemas. Este enfoque dual garante que o alumnado adquira destrezas en ambos ambientes, reflectindo a realidade tecnolóxica de moitas organizacións.

O enfoque do curso é eminentemente práctico, centrado na metodoloxía de "aprender facendo". A finalidade non é a aprendizaxe teórica da programación, senón a súa aplicación directa á resolución de problemas reais da administración de sistemas. A elección de Python como linguaxe principal baséase na súa sinxeleza sintáctica e na súa "curva de aprendizaxe moi plana", o que permite ao alumnado ser produtivo nun período de tempo moi reducido.

A través dunha serie de unidades didácticas coidadosamente secuenciadas, o alumnado construírá o seu coñecemento desde os fundamentos ata a aplicación práctica, culminando na capacidade de desenvolver solucións de automatización para contornos GNU/Linux.

2. Obxectivos da Materia e Contribución ao Desenvolvemento das Competencias Clave

Este apartado define os logros de aprendizaxe que se esperan do alumnado ao finalizar o curso (obxectivos) e detalla como a consecución destes contribúe ao desenvolvemento das competencias clave, esenciais para o seu perfil profesional e a súa aprendizaxe permanente. Os obxectivos xerais da materia son os seguintes:

- **OBX1:** Comprender os conceptos fundamentais da programación e a sintaxe da linguaxe Python, preparando unha contorna de traballo funcional e segura.
- **OBX2:** Desenvolver habilidades para a descomposición de problemas complexos en compoñentes máis sinxelos e deseñar solucións programáticas modulares e eficientes.
- **OBX3:** Aplicar os coñecementos de Python na automatización e xestión de tarefas comúns



da administración de sistemas en contornos GNU/Linux, interactuando co sistema operativo, servizos e bases de datos.

A seguinte táboa establece a relación entre os obxectivos da materia e os descritores das competencias clave asociadas:

Obxectivos da materia	Competencias Clave asociadas (Descritores)
OBX1. Comprender os conceptos fundamentais da programación e a sintaxe da linguaxe Python.	STEM, CD, CPSAA
OBX2. Desenvolver habilidades para a descomposición de problemas e o deseño de solucións.	STEM, CD, CPSAA, CE
OBX3. Aplicar Python na automatización de tarefas de administración de sistemas en GNU/Linux.	CCL, STEM, CD, CE

A estrutura do curso, organizada en unidades didácticas, está deseñada para garantir a consecución progresiva destes obxectivos.

3. Relación de Unidades Didácticas, Secuenciación e Temporización

O curso estrutúrase nun total de 25 sesións, distribuídas en cinco unidades didácticas de 5 sesións cada unha. Esta secuenciación foi deseñada para asegurar unha progresión lóxica e coherente, partindo dos conceptos fundamentais da linguaxe Python e culminando na súa aplicación práctica e avanzada en tarefas de automatización de sistemas.

Unidade 1: Introducción a Python e o Entorno de Traballo

- **Temporización:** 5 Sesións
- **Descrición:** Configuración da contorna de desenvolvemento e primeira aproximación á sintaxe e características de Python.
- **Contidos:**
 - Que é Python?: Historia, evolución e características clave (linguaxe interpretada, de scripting, curva de aprendizaxe).
 - Preparación do Entorno: Instalación de Python, concepto e uso de contornos virtuais.
 - Shell Interactiva: Uso de Python e IPython para experimentación.
 - Primeiro Programa: Sintaxe básica, variables, tipos de datos primitivos e comentarios.
- **Resultados de Aprendizaxe (RAs):** OBX1
- **Criterios de Avaliación (CAs):**
 - CA1.1. Configurar un contorno de desenvolvemento Python, incluíndo a creación e xestión de contornos virtuais.
 - CA1.2. Utilizar a shell interactiva, especialmente IPython, para executar comandos



básicos, declarar variables e realizar operacións sinxelas.

- CA1.3. Escribir e executar un script básico que demostre o uso de variables e a función `print()`.

Unidade 2: Fundamentos da Programación: Control de Fluxo e Funcións

- **Temporización:** 5 Sesións
- **Descrición:** Adquisición das estruturas lóxicas esenciais para a construción de algoritmos e a modularización do código.
- **Contidos:**
 - Operadores: Aritméticos, de comparación e lóxicos.
 - Estruturas de Control de Fluxo: Condicionais (`if-elif-else`) e bucles (`for`, `while`).
 - Funcións: Definición (`def`), paso de argumentos, valores de retorno e alcance de variables.
 - Módulos: Importación e uso de módulos da biblioteca estándar.
- **Resultados de Aprendizaxe (RAs):** OBX1, OBX2
- **Criterios de Avaliación (CAs):**
 - CA2.1. Desenvolver scripts que utilicen estruturas de control de fluxo (condicionais e bucles) para resolver problemas sinxelos de decisión e iteración.
 - CA2.2. Crear funcións modulares que encapsulen lóxica reutilizable, aceptando parámetros e devolvendo resultados.
 - CA2.3. Corrixir fragmentos de código con erros de sintaxe ou de lóxica elemental.

Unidade 3: Estruturas de Datos Esenciais

- **Temporización:** 5 Sesións
- **Descrición:** Manexo das coleccións de datos fundamentais en Python para a xestión de información estruturada.
- **Contidos:**
 - Listas: Creación, acceso, modificación e iteración.
 - Tuplas: Concepto de inmutabilidade e casos de uso.
 - Dicionarios: Estruturas clave-valor para xestionar datos e configuracións.
 - Strings Avanzadas: Métodos de manipulación e formato con f-strings.
- **Resultados de Aprendizaxe (RAs):** OBX1, OBX2
- **Criterios de Avaliación (CAs):**
 - CA3.1. Utilizar listas e dicionarios para almacenar e manipular conxuntos de datos estruturados, como inventarios de servizos ou usuarios.
 - CA3.2. Implementar un script que xestione unha colección de recursos do sistema (p. ex. asignación de IPs) mediante o uso de dicionarios e funcións.
 - CA3.3. Procesar cadeas de texto para extraer información relevante, utilizando métodos de string e f-strings para xerar saídas formatadas.



Unidade 4: Programación Orientada a Obxectos (POO) e Manexo de Ficheiros

- **Temporización:** 5 Sesións
- **Descrición:** Introducción ao paradigma orientado a obxectos para modelar entidades do mundo real e persistir información en ficheiros.
- **Contidos:**
 - Introducción á POO: Conceptos de clase, obxecto, atributo e método.
 - Definición de Clases: O construtor `__init__`, o punteiro `self` e a representación de obxectos con `__str__`.
 - Manexo de Ficheiros: Lectura e escritura de ficheiros de texto co xestor de contexto `with`.
 - Xestión de Erros: Uso básico de `try-except`.
- **Resultados de Aprendizaxe (RAs):** OBX1, OBX2
- **Criterios de Avaliación (CAs):**
 - CA4.1. Definir e instanciar clases sinxelas en Python para modelar entidades do sistema (p. ex., un servidor, un usuario).
 - CA4.2. Escribir un script que lea datos dun ficheiro de texto, os procese e garde os resultados noutro ficheiro.
 - CA4.3. Utilizar o xestor de contexto `with` para garantir o peche seguro dos recursos de ficheiros.

Unidade 5: Python para a Automatización de Sistemas GNU/Linux

- **Temporización:** 5 Sesións
- **Descrición:** Aplicación final dos coñecementos adquiridos para automatizar tarefas reais de administración en GNU/Linux.
- **Contidos:**
 - Módulos `os` e `sys`: Interacción co sistema de ficheiros e variables de contorna.
 - Execución de Comandos Externos: Uso do módulo `subprocess` para executar e capturar a saída de comandos do sistema.
 - Interacción con Bases de Datos: Conexión e execución de consultas SQL sobre MariaDB.
- **Resultados de Aprendizaxe (RAs):** OBX2, OBX3
- **Criterios de Avaliación (CAs):**
 - CA5.1. Automatizar tarefas comúns de administración de sistemas en GNU/Linux (p. ex. xestión de ficheiros, consulta de estado de servizos) mediante a execución de comandos externos co módulo `subprocess`.
 - CA5.2. Realizar operacións básicas de base de datos (creación, inserción e consulta) nunha base de datos MariaDB usando Python, empregando consultas parametrizadas para garantir a seguridade das operacións.
 - CA5.3. Desenvolver un script de automatización completo que integre a interacción co sistema de ficheiros e a execución de comandos externos para xerar un informe.



A metodoloxía empregada para impartir estas unidades didácticas está deseñada para maximizar a aprendizaxe práctica e a súa aplicabilidade directa.

4. Metodoloxía

O enfoque principal estará na resolución de problemas reais e na simulación de escenarios comúns na administración de sistemas, co fin de garantir unha transferencia directa do coñecemento adquirido ao ámbito profesional do alumnado.

As **concrecións metodolóxicas** organizaranse do seguinte xeito:

- **Estratexias:**
 - **Uso de Jupyter Notebooks:** Empregaranse como ferramenta principal para combinar a exposición de conceptos teóricos con exemplos de código executable. Isto permitirá ao alumnado experimentar de forma interactiva e consolidar a aprendizaxe de maneira guiada.
 - **Desenvolvemento de pequenos proxectos:** Cada unidade culminará con tarefas prácticas que, en conxunto, construírán pequenas solucións de automatización. Este enfoque orientado a proxectos fomenta a integración de coñecementos e a visión global sobre a resolución de problemas.
- **Agrupamentos:**
 - Combinarase o **traballo individual**, fundamental para a asimilación dos conceptos básicos de programación e a sintaxe da linguaxe, co **traballo en parellas ou pequenos grupos**. Este último aplicarase na resolución de problemas máis complexos ou pequenos proxectos, fomentando a colaboración, a comunicación e a aprendizaxe entre iguais.
- **Materiais e Recursos Didácticos:**
 - **Software:**
 - Máquina virtual con sistema operativo Windows para a instalación inicial de ferramentas.
 - Distribución GNU/Linux como contorno principal para as tarefas de administración.
 - Intérprete de Python 3.x.
 - **Visual Studio Code (VS Code)** como editor de código principal, aproveitando as súas capacidades de completión de código, depuración e xestión de engadidos.
 - **Jupyter Notebooks** para a práctica interactiva.
 - Sistema xestor de bases de datos **MariaDB**, despregado a través de **Docker** para simplificar a súa instalación e configuración.
 - **Documentación:**
 - Cadernos de traballo dixitais (Jupyter Notebooks) proporcionados polo docente.
 - Acceso á documentación oficial de Python e das librarías utilizadas, fomentando a autonomía na busca de información técnica.



Esta metodoloxía activa e baseada en ferramentas profesionais está directamente aliñada co sistema de avaliación, que primará a aplicación práctica dos coñecementos.

5. Avaliación

A avaliación do alumnado será continua e formativa, deseñada para medir non só a adquisición de coñecementos teóricos, senón, fundamentalmente, a súa capacidade para aplicar a programación na resolución de problemas prácticos de administración de sistemas.

5.1. Avaliación Inicial

Ao comezo do curso, realizarase unha proba práctica non cualificable para valorar os coñecementos previos do alumnado sobre o manexo básico da liña de comandos, tanto en contornos Windows como GNU/Linux. O resultado desta proba permitirá axustar o punto de partida das primeiras sesións e identificar necesidades de reforzo iniciais.

5.2. Criterios de Cualificación

O sistema de cualificación reflicte o enfoque eminentemente práctico da materia, ponderando os instrumentos de avaliación do seguinte xeito:

- **Probas prácticas e exercicios (30%):** Este bloque corresponde á avaliación continua dos scripts, notebooks e pequenos proxectos de automatización realizados ao longo das cinco unidades didácticas. Valorarase a funcionalidade do código, a súa claridade, a eficiencia na resolución do problema proposto e a aplicación de boas prácticas (uso de contornos virtuais, funcións, etc.).
- **Proba teórico-práctica final (70%):** Ao remate do curso, realizarase unha proba individual que combinará preguntas conceptuais sobre os fundamentos de Python coa resolución de exercicios de codificación curtos. O seu obxectivo é asegurar a comprensión global dos conceptos clave impartidos.

O grao mínimo de consecución para superar a materia é obter unha cualificación final igual ou superior a 5 sobre 10, sendo requisito indispensable obter unha cualificación mínima de 4 sobre 10 en cada unha das partes ponderadas para poder facer media.

5.3. Criterios de Recuperación

O alumnado que non supere a materia a través da avaliación continua terá dereito a realizar unha proba de recuperación final. Esta proba consistirá na resolución dun caso práctico de automatización de sistemas que abranguerá e integrará os contidos máis relevantes de todas as unidades didácticas.

A planificación da avaliación complétase coas medidas de atención á diversidade, que garanten unha resposta educativa axustada ás necesidades de todo o alumnado.



6. Medidas de Atención á Diversidade

Para garantir que todo o alumnado poida alcanzar os obxectivos formativos, aplicaranse medidas de atención á diversidade de carácter ordinario. Para o alumnado que presente maiores dificultades na asimilación dos conceptos de programación, propoñeranse **exercicios de reforzo** guiados e con pautas adicionais. Pola contra, para aquel alumnado que demostre unha maior capacidade e ritmo de aprendizaxe, ofreceranse **retos de ampliación** que consistirán en problemas de maior complexidade ou na exploración de módulos adicionais de Python. A metodoloxía práctica do curso facilita a **tutorización individualizada** durante as sesións, permitindo ao docente ofrecer apoio específico e adaptado ás necesidades detectadas en cada estudante.

7. Concreción dos Elementos Transversais

Ao longo do curso traballarase de forma integrada diversos elementos transversais. A propia natureza da materia fomenta o desenvolvemento da **competencia dixital** a través do uso de ferramentas profesionais. O **pensamento computacional e a resolución de problemas** son o eixe central de todas as actividades propostas. A necesidade de consultar documentación técnica en liña para resolver dúbidas ou ampliar coñecementos promoverá a **comprensión lectora** de textos especializados. Finalmente, o enfoque práctico e a resolución de retos fomentarán a **autonomía e a aprendizaxe continua**, capacidades esenciais para un profesional do sector tecnolóxico.

8. Avaliación da Práctica Docente

Ao finalizar o curso, levarase a cabo unha avaliación da práctica docente e da propia programación didáctica. Este proceso realizarase a través de dous instrumentos principais: a análise obxectiva dos resultados académicos do alumnado, que permitirá valorar o grao de consecución dos obxectivos, e un cuestionario anónimo que recollerá as impresións, suxestións e valoracións do alumnado sobre a metodoloxía, os recursos e o desenvolvemento do curso. O obxectivo final é identificar os puntos fortes e as áreas de mellora para axustar e optimizar esta programación de cara a futuros cursos académicos.