CURSO 2023-2024
CICLO SUPERIOR DE DESARROLLO DE APLICACIONES WEB
IES AGUADULCE

Programación

Francisco Javier Sueza Rodríguez

26 de septiembre de 2023

Índice general

Índice de figuras

Tema 1

Introducción a la Programación

En esta primera unidad vamos a estudiar los conceptos básicos de la programación de aplicaciones. Comenzaremos estudiando que es la programación, que técnicas podemos emplear, que herramientas podemos utilizar y cual es objetivo que pretendemos alcanzar. Analizaremos las diferentes paradigmas de programación existentes, identificaremos las fases del desarrollo de un programa.

Una vez realizada una introducción general, detallaremos las características relevantes de los principales lenguajes de programación, para a continuación centrarnos en el lenguaje que vamos a usar durante toda esta asignatura, **Java**, dando a conocer también que herramientas podemos usar para que nuestro desarrollo sea más sencillo con este lenguaje.

1.1 Programas: Buscando una Solución

La principal razón que mueve a una persona hacia el aprendizaje de la programación es utilizar el ordenador como una herramienta para resolver diferentes problemas. Al igual que en la vida real, las búsqueda y obtención de una solución requiere de una serie de **pasos fundamentales**.

En la vida real...	En Programación...
Observación de la situación o problema.	**Análisis del problema:** requiere que el problema sea definido y comprendido claramente para que pueda ser analizado con todo detalle.
Pensamos en una o varias posibles soluciones.	**Diseño o desarrollo de algoritmos:** procedimiento paso a paso para solucionar el problema dado.
Aplicamos la solución que estimamos más adecuada.	**Resolución del algoritmo elegido en la computadora:** consiste en convertir el algoritmo en programa, ejecutarlo y comprobar que soluciona verdaderamente el problema.

Figura 1.1.1: Pasos para la resolución de un problema

Para que una **solución** se considere **correcta** tiene que tener principalmente dos características:

- **Corrección y Eficacia**: si resuelve el problema de forma adecuada.
- **Eficiencia**: si lo realiza en un tiempo mínimo y con un uso óptimo de los recursos del sistema.

Para construir esta solución, hay que tener en cuenta algunos conceptos ligados a la programación, como son:

1. **Abstracción**: se trata de realizar una análisis del problema para descomponerlo en problemas más pequeños y de menos complejidad de manera precisa. **Divide y Vencerás**: es una filosofía general para resolver problemas y de aquí que no solo forme parte del vocabulario informático, sino que también se utiliza en otros muchos ámbitos.
2. **Encapsulación**: consiste en ocultar la información de un objeto o función de forma que se pueda implementar de diferentes maneras sin que esto afecte al resto de objetos.
3. **Modularidad**: estructuraremos cada parte en módulos independientes, cada uno con su función correspondiente.

Todo estos conceptos, deberemos tenerlos en cuenta a la hora de analizar el problema, para llegar a un solución lo más óptima posible.

1.1.1 Algoritmos y Programas

Una vez realizado el análisis del problema, tenemos que diseñar y desarrollar un **algoritmo** adecuado que pueda solucionarlo. Pero, ¿que es un algoritmo?

Un **algoritmo** es una secuencia ordenada de pasos, descrita sin ambigüedades, que conducen a la solución de un problema.

Los algoritmos deben ser **independientes** de los **lenguajes de programación** y de las **computadoras** donde se ejecutan, de forma que puedan implementarse sobre cualquier ordenador empleando cualquier lenguaje de programación. Esto facilita que una misma solución pueda emplearse para el mismo problema en diferentes dispositivos.

La **diferencia** entre un algoritmo y un **programa** es que en este último lo pasos deben escribirse en un **lenguaje de programación concreto** para que pueda ser ejecutado en el ordenador y así obtener la solución deseada.

Los **lenguajes de programación** son solo un medio para expresar el algoritmo y el ordenador un procesador para ejecutarlo. El diseño de algoritmos es una tare que requiere de la creatividad y conocimientos de las técnicas de programación del programador, así, diferentes programadores pueden desarrollar diferentes algoritmos para resolver un mismo problema.

Las principales **características** que debe cumplir un **algoritmo** son:

- Debe ser **preciso** e indicar el orden en el que se realiza cada paso.
- Debe estar **bien definido**, si se ejecuta dos o más veces, debemos obtener el mismo resultado.
- Debe ser **finito**, teniendo un número de pasos bien determinado.

Cuando los problemas complejos, debemos descomponer estos en subproblemas más simples, y estos a su vez en otros más pequeños. Es lo que se conoce como **diseño descendente** o **diseño modular** y se basa en el lema **Divide y Vencerás**.

Para **representar gráficamente** los algoritmos tenemos diferentes herramientas que nos ayudarán a describir su comportamiento de una forma precisa y genérica, que nos facilitará la implementación del algoritmo en diferentes lenguajes de programación. Las principales herramientas que tenemos son:

- **Diagramas de Flujo**: esta técnica utiliza símbolos gráficos para representar el flujo de ejecución del algoritmo y suelen ser empleados en la fase de análisis.

- **Pseudocódigo**: se basa en el uso de palabras clave en lenguaje natural, representando las constantes, variables y otras estructuras de programación de forma escrita. Es la técnica mas utilizada actualmente.
- **Tablas de Decisión**: es un tabla que representa las diferentes condiciones del problema con sus respectivas acciones. Suele ser una técnica de apoyo a pseudocódigo cuando existen circunstancias condicionales complejas.

1.2 Fases de la Programación

Sea cual sea el estilo que escojamos para resolver el problema, deberemos realizar el proceso aplicando un método a nuestro trabajo. Así, el **proceso de creación de software** se puede dividir en las siguientes **fases**:

- **Fase de resolución del problema**
- **Fase de implementación**
- **Fase de explotación y mantenimiento**

En los siguientes puntos, analizaremos cada una de estas fases.

1.2.1 Resolución del Problema

Esta es la primera fase del desarrollo del programa, para la cuál deberá estar bien definido cual es el problema que se quiere solucionar y tener un comprensión clara de éste para poder realizar su análisis y diseño. Esta fase se puede dividir en **dos etapas**:

1. **Etapa de Análisis**:

   En esta primera etapa se debe analizar el problema, lo que nos va a indicar las especificaciones y requisitos que la aplicación debe cumplir. Para llevar esto acabo, se deberán realizar diferentes entrevistas entre el programador y el cliente/usuario para precisar cuales son las características que debe tener la aplicación, especificando, entre otras cosas, los procesos y estructuras que deberá tener ésta. La creación de **prototipos** será muy útil en esta fase para saber con mayor exactitud los puntos a tratar.

   Esta etapa proporcionará una idea general de lo que se solicita, realizando sucesivos refinamientos posteriormente que servirá para determinar cual es la información que ofrecerá la resolución del problema y que datos son necesarios para resolver este.

2. **Etapa de Diseño**:

   En esta tapa se convierte la especificación de la etapa de análisis en un diseño más detallado que define el comportamiento o la secuencia lógica de instrucciones capaz de resolver el problema planteado. Estos pasos sucesivos, constituyen lo que ya hemos definido como algoritmo.

   Antes de pasar a la implementación del algoritmo, tenemos que tener claro que la solución que se propone es la adecuada. Para ello, toda solución necesitará de la **prueba** o **traza** del programa. Este procedimiento consistirá en el seguimiento paso a paso de cada instrucción del algoritmo utilizando los datos correctos. Si la solución aportada contiene errores, deberemos volver a la etapa de análisis, si no, podremos pasar a la fase de implementación.

Bibliografía