

Programació 1

GRAU en EI

Introducción a la algorítmica

CCIA - DIEI - UdL

¿ Qué es un algoritmo ?

Un algoritmo es la descripción detallada y precisa de los pasos a seguir para llevar a cabo un cierto proceso

Ejemplos de procesos:

- Freír un huevo frito
- Poner la lavadora
- Sacar dinero de un cajero
- Programar el video para grabar un programa

Componentes de un proceso

Elementos de entrada: Elementos que son necesarios para poder iniciar el proceso

Objetivo del proceso: Que efecto tendrá el proceso sobre los elementos de entrada, o que nuevos elementos podrá producir, **si se ejecuta con éxito**

Descripción del proceso: dos posibilidades

- Descripción informal (con lenguaje natural)
- Descripción formal (con lenguaje algorítmico)

El proceso de ligar en una fiesta

Supongamos que tenemos un “robot” con la intención de ligar en una fiesta y que un ordenador controla sus acciones

El robot quiere ligar con una persona. Sus sensores le permiten captar el estado emocional de la persona

Actividades básicas que el robot sabe realizar:

- 1) Hablar con personas
- 2) Invitarlas a beber

El proceso de ligar en una fiesta

Estados en los cuales podrá encontrarse una persona:

pasa de nosotros (1)

nos escucha (2)

se "arrima" (3)

entrega total (4)

Bebidas posibles:

agua (1)

vodka (2)

ron (3)

malibú (4)

Elementos de entrada: estado inicial de la persona objetivo y su bebida favorita

Objetivo: el proceso deberá cambiar el estado de la persona objetivo a **entrega total**

El proceso de ligar en una fiesta

Descripción **informal** del proceso:

Si la persona pasa de nosotros, entablar conversación inicial con lo que pasará a escucharnos.

A continuación, mientras la persona no se entregue a nosotros, ir invitándola a copas de su bebida favorita al mismo tiempo que reanudamos la conversación

El proceso de ligar en una fiesta

Descripción **formal** del proceso (**algoritmo**):

Algoritmo ligar_en_una_fiesta **es**

const estado_inicial = 1; bebida_favorita = 4 **fconst**

var estado_persona: entero **fvar**

estado_persona := estado_inicial;

si (estado_persona = 1) **entonces**

 estado_persona := entablar_conversación_inicial();

fsi

mientras (estado_persona \neq 4) **hacer**

 estado_persona := conversar_e_invitar(bebida_favorita);

fmientras

falgoritmo

El proceso de ligar en una fiesta

¿ De qué depende que el algoritmo funcione ?

1) Todos los pasos que lleva a cabo deben realizarse en el orden en el que aparecen

1) Las llamadas realizadas a las dos acciones deben finalizar en algún momento y devolver el estado de la persona

3) Dentro de la composición iterativa, alguna de las llamadas realizadas a la acción `conversar_e_invitar` deberá devolver como estado de la persona "entrega total"

El proceso de ligar en una fiesta

Para poder **implementar** el algoritmo debemos:

- Poder pedir al robot que ejecute las acciones

1) entablar_conversación_inicial()

2) conversar_e_invitar(bebida_favorita)

de forma que lleven a cabo **lo que se espera de ellas** y que después de su ejecución el robot pueda **obtener el estado** en el que queda la persona.

- Traducir el algoritmo a un lenguaje que entienda el ordenador del robot

Implementación y ejecución

La **implementación** de un algoritmo consiste en traducir un algoritmo a un **lenguaje de programación** que entienda el ordenador



Tras la implementación, podremos **ejecutar** el programa en el ordenador para obtener los resultados deseados

Importante:

Un algoritmo sólo puede ser implementado si los elementos con los que trabaja pueden ser representados en la memoria del ordenador como datos **constantes** o **variables**

El proceso de ligar en una fiesta

Componentes del algoritmo

Declaración de constantes:

```
const estado_inicial = 1; bebida_favorita = 4 fconst
```

Declara dos constantes, que son dos palabras que quedan asociadas a dos números enteros.

Declaración de variables:

```
var estado_persona: entero fvar
```

Declara la creación de una **variable**, llamada estado_persona. Esta variable actuará como una caja donde el algoritmo podrá guardar un número **entero**. El algoritmo podrá cambiar el número guardado en la variable cuantas veces quiera, pero en la variable sólo se encontrará el último valor guardado.

El proceso de ligar en una fiesta

Componentes del algoritmo

Asignación inicial:

```
estado_persona := estado_inicial;
```

Cambia el valor de la **variable** estado_persona al valor que indica la constante estado_inicial

Composición alternativa:

```
si (estado_persona = 1) entonces  
    estado_persona := entablar_conversación_inicial();  
fsi
```

Comprueba si el estado de la persona es igual a “pasa de nosotros”, en cuyo caso llama a la acción que se encarga de entablar la conversación inicial, y que después deberá devolvernos el estado en el que queda la persona, para asignarlo a nuestra variable.

El proceso de ligar en una fiesta

Componentes del algoritmo

Composición iterativa:

```
mientras (estado_persona  $\neq$  4 ) hacer  
    estado_persona := conversar_e_invitar( bebida_favorita );  
fmientras
```

Repite una misma actividad mientras se cumpla cierta condición. En este caso, la actividad es la acción `conversar_e_invitar`, que necesita como información de entrada la bebida favorita. La condición es que el estado de la persona no sea "entrega total"

Esta composición iterativa sólo finalizará cuando una llamada a la acción `conversar_e_invitar` consiga dejar el estado de la persona al valor "entrega total", ya que el estado devuelto por la llamada a la acción se asigna a la variable `estado_persona`

¿ Qué debemos aprender en Programación?

Utilizar las estructuras algorítmicas y tipos de datos básicos

Analizar y diseñar algoritmos para problemas de gestión de secuencias

Aplicar las estrategias adecuadas al diseño de algoritmos en base a las diferentes tipologías de problemas.

Utilizar las estructuras de datos adecuadas para recoger y almacenar la información que procesan los programas

Implementar los algoritmos en un lenguaje de alto nivel.
Este lenguaje será el **C y C++**

Utilizar las técnicas de gestión de memoria dinámica en C