

# Programación imperativa

Ricardo Pérez López

IES Doñana, curso 2019/2020



1. Modelo de ejecución
2. Cambios de estado explícitos
3. Efectos laterales
4. Saltos
5. Los *arrays* como estructura de datos mutable básica

# 1. Modelo de ejecución

1.1 Máquina de estados

1.2 Secuencia de instrucciones

## 1.1. Máquina de estados

## 1.2. Secuencia de instrucciones

## 2. Cambios de estado explícitos

2.1 Celdas

2.2 Asignación destructiva (o asignación múltiple)

2.3 Asignación por referencia

## 2.1. Celdas

## 2.2. Asignación destructiva (o asignación múltiple)



## 2.3. Asignación por referencia

## 3. Efectos laterales

3.1 Transparencia referencial

3.2 Entrada y salida por consola

## 3.1. Transparencia referencial

## 3.2. Entrada y salida por consola

## 3.2. La sentencia echo

## 3.2. Las funciones `var_dump()` y `print_r()`

## 3.2. Las funciones `fgets()` y `fscanf()`

## 4. Saltos

4.1 Incondicionales: la sentencia `goto`

4.2 Condicionales: la sentencia `if (...) goto`

4.3 Implementación de bucles mediante saltos condicionales



## 4.1. Incondicionales: la sentencia goto

## 4.2. Condicionales: la sentencia `if (...) goto`

## 4.3. Implementación de bucles mediante saltos condicionales

## 5. Los *arrays* como estructura de datos mutable básica

- 5.1 Creación, acceso y modificación
- 5.2 Recorrido y búsqueda en un *array*
- 5.3 *Arrays* multidimensionales
- 5.4 Funciones de manejo de *arrays*

## 5.1. Creación, acceso y modificación

## 5.2. Recorrido y búsqueda en un *array*

## 5.3. *Arrays* multidimensionales

## 5.4. Funciones de manejo de *arrays*