

# Programación imperativa

Ricardo Pérez López

IES Doñana, curso 2019/2020

## Índice general

<b>1. Modelo de ejecución</b>	<b>2</b>
1.1. Máquina de estados . . . . .	2
1.2. Secuencia de instrucciones . . . . .	2
<b>2. Cambios de estado explícitos</b>	<b>2</b>
2.1. Celdas . . . . .	2
2.2. Asignación destructiva (o asignación múltiple) . . . . .	2
2.3. Asignación por referencia . . . . .	2
<b>3. Efectos laterales</b>	<b>2</b>
3.1. Transparencia referencial . . . . .	2
3.2. Entrada y salida por consola . . . . .	2
<b>4. Saltos</b>	<b>2</b>
4.1. Incondicionales: la sentencia <code>goto</code> . . . . .	2
4.2. Condicionales: la sentencia <code>if (...) goto</code> . . . . .	2
4.3. Implementación de bucles mediante saltos condicionales . . . . .	2
<b>5. Los <i>arrays</i> como estructura de datos mutable básica</b>	<b>2</b>
5.1. Creación, acceso y modificación . . . . .	2
5.2. Recorrido y búsqueda en un <i>array</i> . . . . .	2
5.3. <i>Arrays</i> multidimensionales . . . . .	2
5.4. Funciones de manejo de <i>arrays</i> . . . . .	2

## 1. Modelo de ejecución

### 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

## 3. Efectos laterales

### 3.1. Transparencia referencial

### 3.2. Entrada y salida por consola

#### 3.2.1. La sentencia `echo`

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

#### 3.2.3. 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

## 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*