Redictado Taller de programación 2020

CLASE 1

Ordenación de vectores

TEMAS DE LA CLASE

1. Ejercitación
2. Concepto de Ordenación
3. Método de Ordenación por Inserción
    - *Pseudocódigo*
    - *Análisis de casos*
    - *Análisis teórico*

Ejercitación

ACTIVIDAD 1: Implementar programaVECTORDELISTAS para resolver el siguiente problema:

Se cuenta con información de los empleados de una empresa. De cada empleado se conoce su número de empleado, apellido, año de ingreso a la empresa (1980..2019) y categoría (1..4).

El programa debe:

a. Leer los datos de empleados hasta que se ingresa el nro de empleado 0 y guardarlos ordenados alfabéticamente por apellido y agrupados por categoría.
b. Una vez guardados, mostrar los apellidos y los códigos de los empleados pertenecientes a cada categoría.

Enviar a través de la Mensajería de Ideas, ProgramaVECTORDELISTAS.pas al docente asignado al grupo.

Ejercitación

Se cuenta con información de los empleados de una empresa. De cada empleado se conoce su número de empleado, apellido, año de ingreso a la empresa (1980..2019) y categoría (1..4). El programa debe:

a. Leer los datos de empleados hasta que se ingresa el nro de empleado 0 y **guardarlos ordenados alfabéticamente por apellido y agrupados por categoría.**

b. Una vez guardados, mostrar los apellidos y los códigos de los empleados pertenecientes a cada categoría.

- Declarar tipos: Lista de Empleados; Vector de Listas.

- Inciso a) Cargar la estructura

```
procedure CrearVectorDeListas (VAR v: vecListas)
        Iniciar las listas del vector en nil
        Leer empleados hasta el nro. 0.
                A cada uno, lo inserto en la lista de su categoría (v[categ])
                orden por apellido
```

Reuso: InsertarEnLista (clase 0)

Ejercitación

Se cuenta con información de los empleados de una empresa. De cada empleado se conoce su número de empleado, apellido, año de ingreso a la empresa (1980..2019) y categoría (1..4). El programa debe:

a. Leer los datos de empleados hasta que se ingresa el nro de empleado 0 y guardarlos ordenados alfabéticamente por apellido y agrupados por categoría.

b. Una vez guardados, **mostrar** los apellidos y los códigos **de los empleados pertenecientes a cada categoría.**

- Inciso b) Mostrar la estructura

```
procedure ImprimirVectorDeListas (v: vecListas)
        Para cada categoría (i)
                Imprimo la lista de esa categoría (v[i])
```

Reuso: ImprimirLista (clase 0)

Ejercitación

ACTIVIDAD 2: Implementar programaPELICULAS para resolver el siguiente problema:

Un cine posee la lista de películas que proyectará durante el mes de Febrero. De cada película se tiene: código de película, título de la película, código de género (1: acción, 2: aventura, 3: drama, 4: suspenso, 5: comedia, 6: bélica, 7: documental y 8: terror) y puntaje promedio otorgado por las críticas.
Escribir un programa que:

a. Lea los datos de películas y los almacene ordenados por código de película y agrupados por código de género. La lectura finaliza cuando se lee el código de película -1.
b. Una vez almacenada la información, informe el código de género que más puntaje obtuvo entre todas las críticas.
c. A partir de un código de película y un código de género que se leen, elimine de ser posible dicha película.

CONCEPTO DE ORDENACIÓN

2. CONCEPTO DE ORDENACIÓN.

Algoritmo de Ordenación

Proceso por el cual, un grupo de elementos puede ser ordenado.

¿Por qué es importante una operación de ordenación en arreglos?

Analicemos distintas situaciones problemáticas donde ésta operación es necesaria…

2. CONCEPTO DE ORDENACIÓN.

MÉTODOS DE ORDENACIÓN CLÁSICOS

- Selección
- Intercambio
- Inserción

3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

Se parte de una secuencia de dos ítems y se ordena.

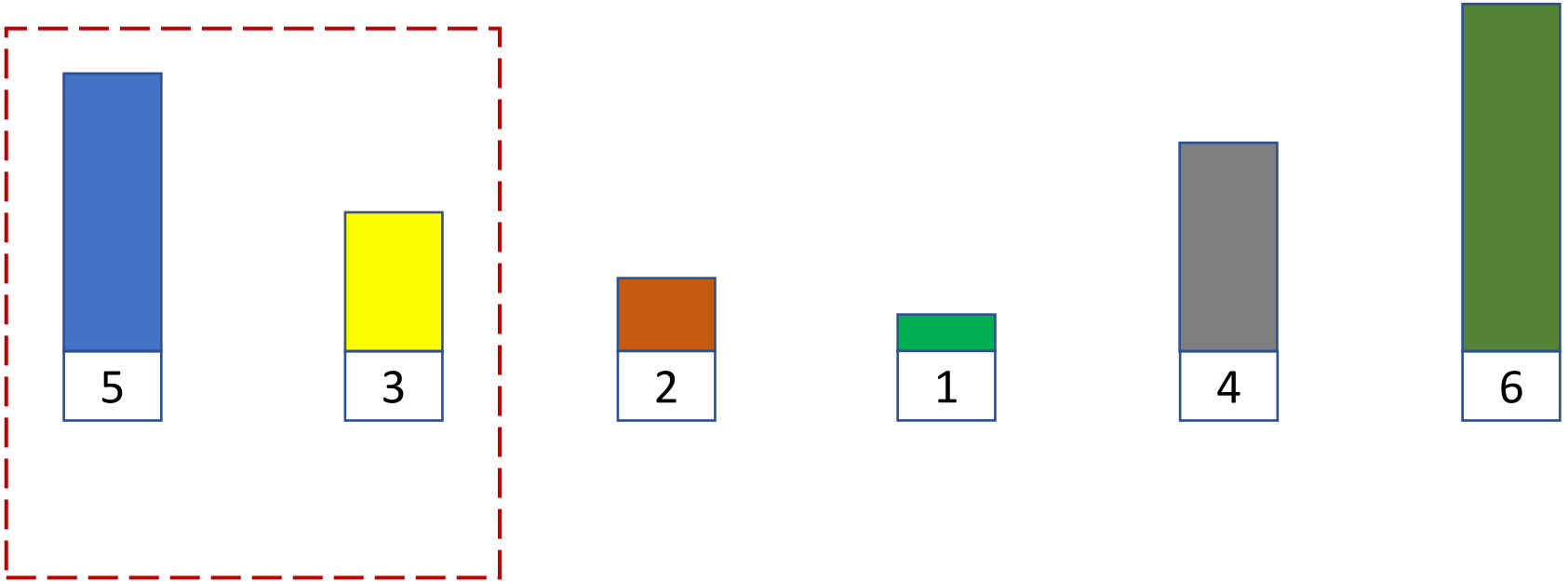
En cada pasada se "*agrega*" un ítem y se inserta en la posición correspondiente en el arreglo ordenado.

Veamos el ejemplo…

3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…

i=2 elemento a ordenar = 3

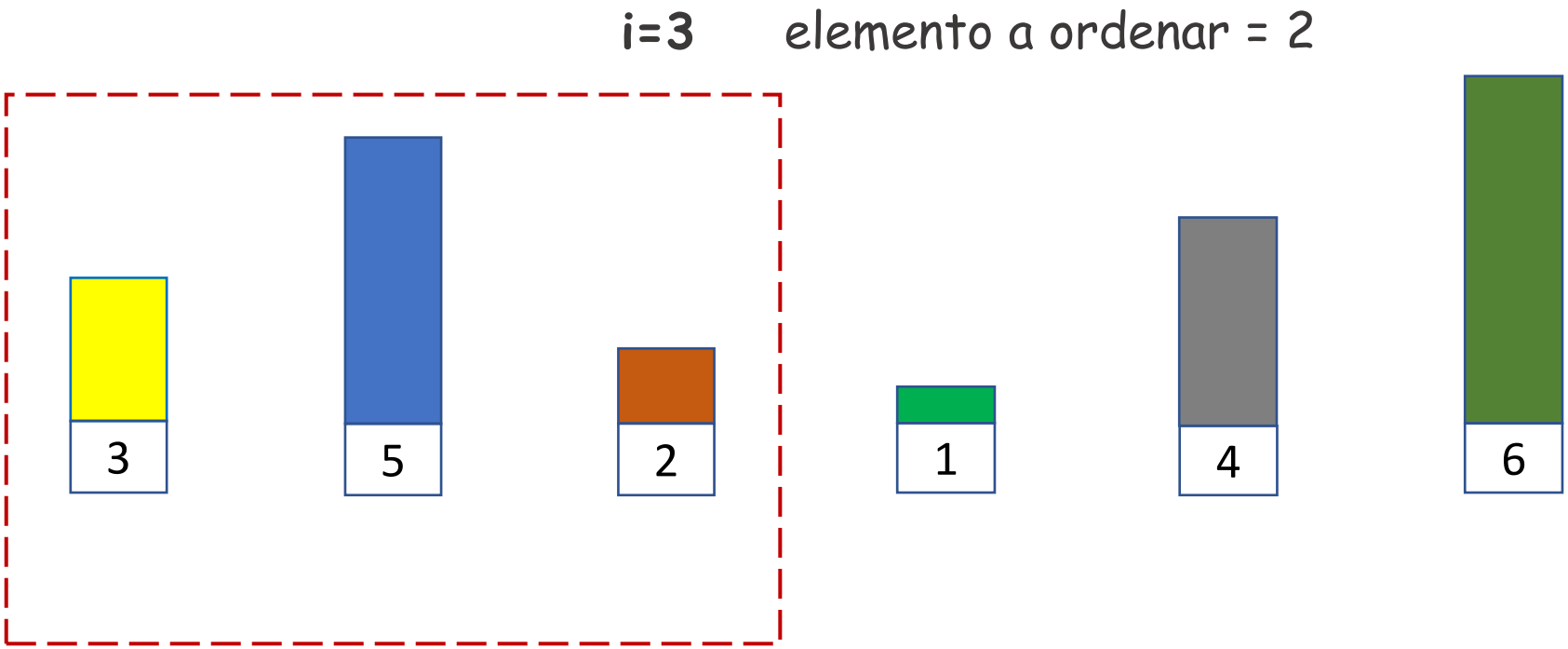


3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…

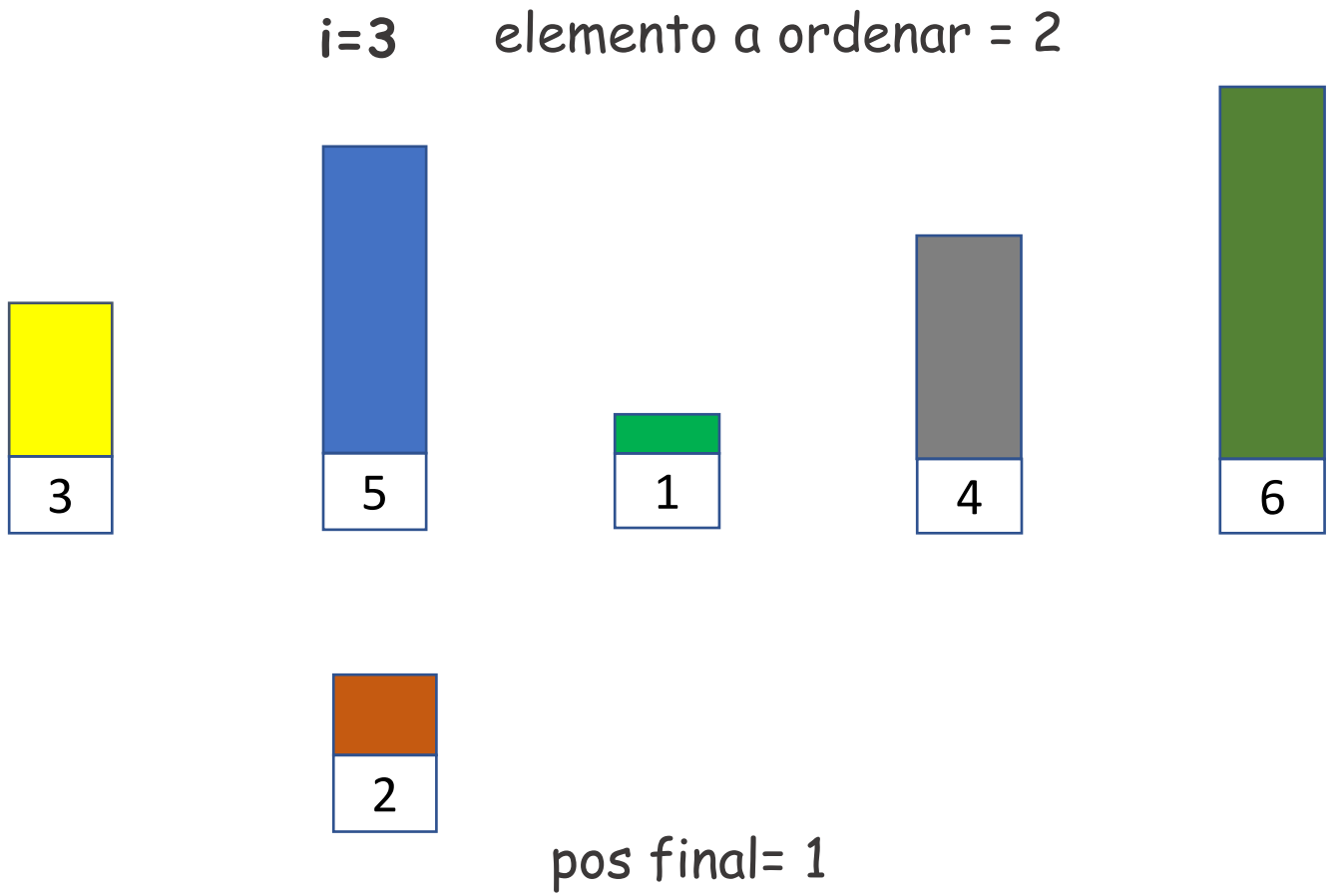
3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…



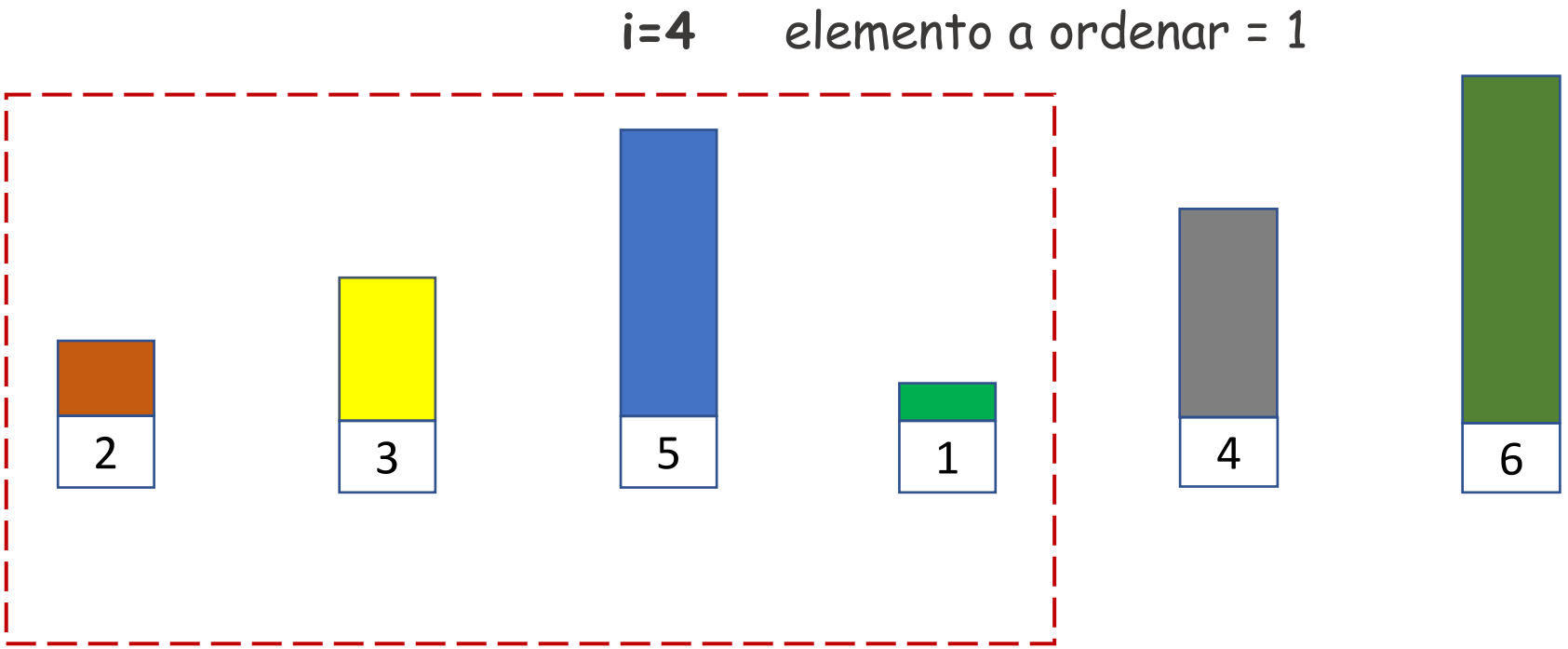
3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…



3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

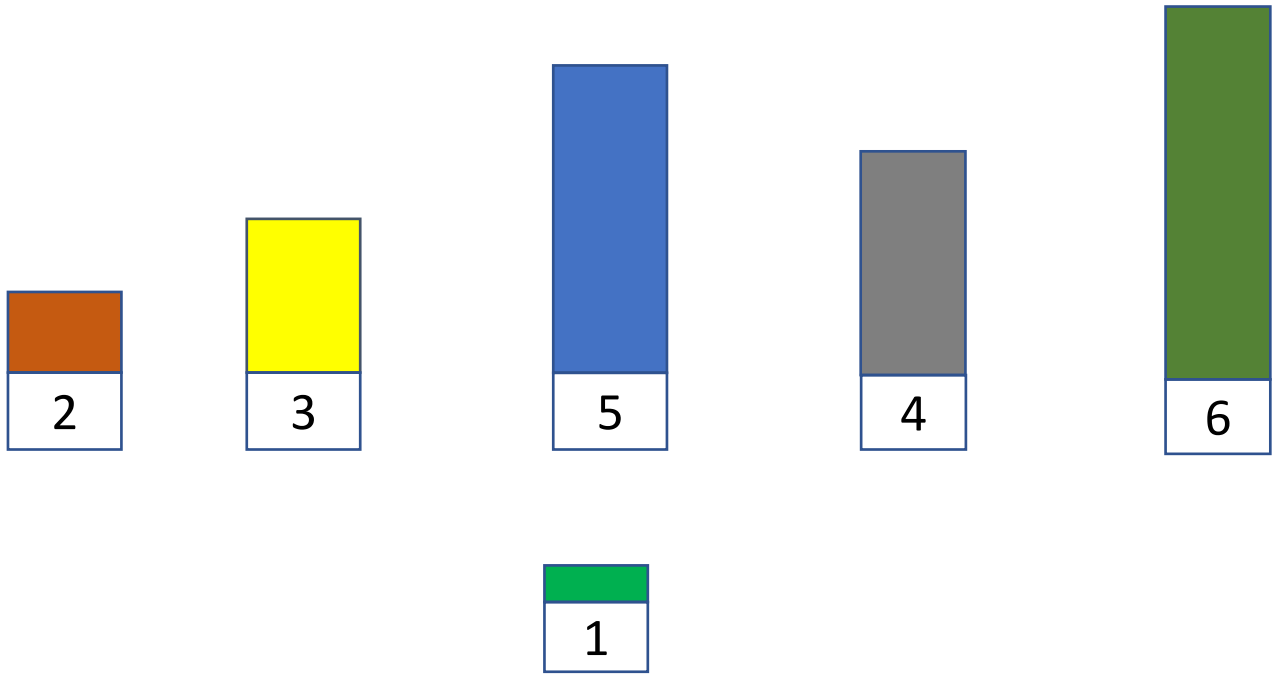
EJEMPLO: Ordenando de menor a mayor…



3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…

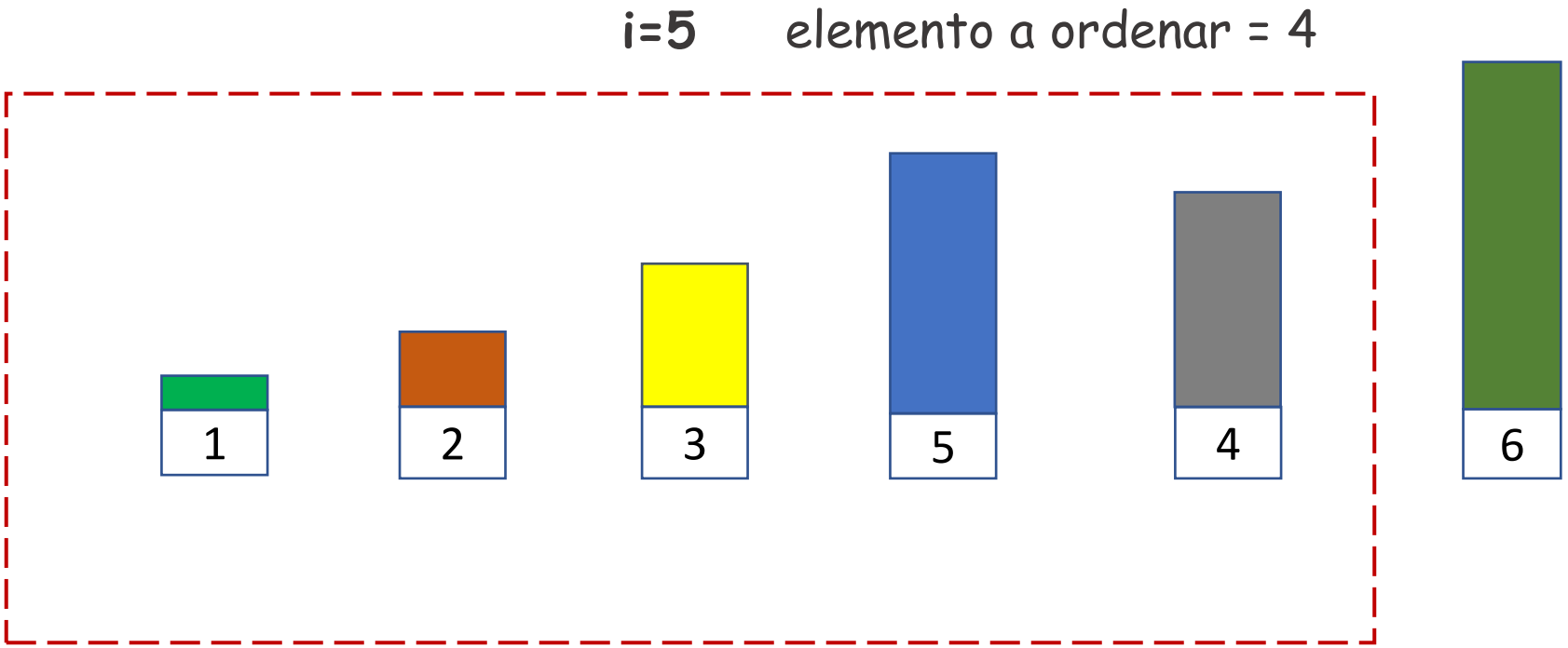
i=4 elemento a ordenar = 1



pos final= 1

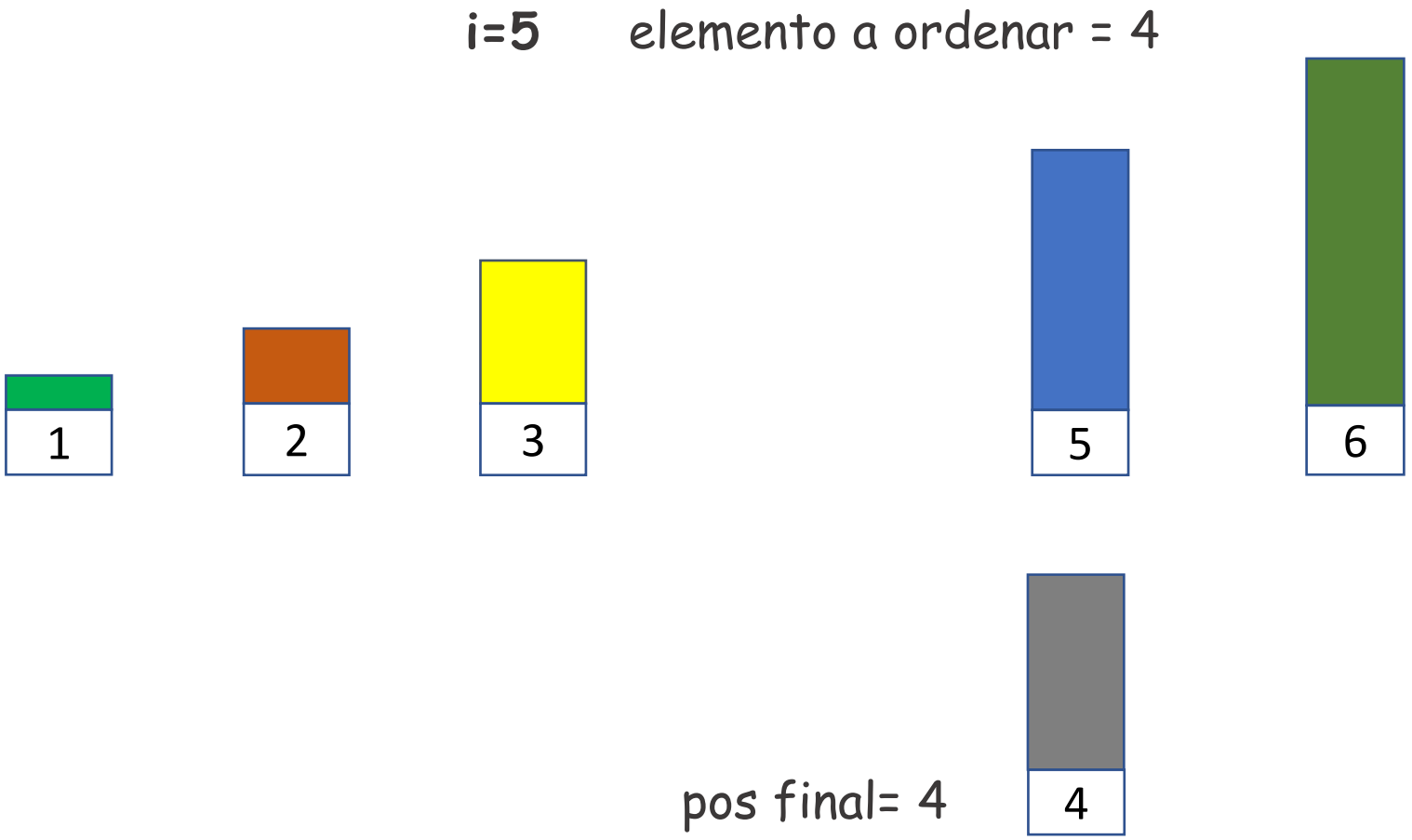
3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…



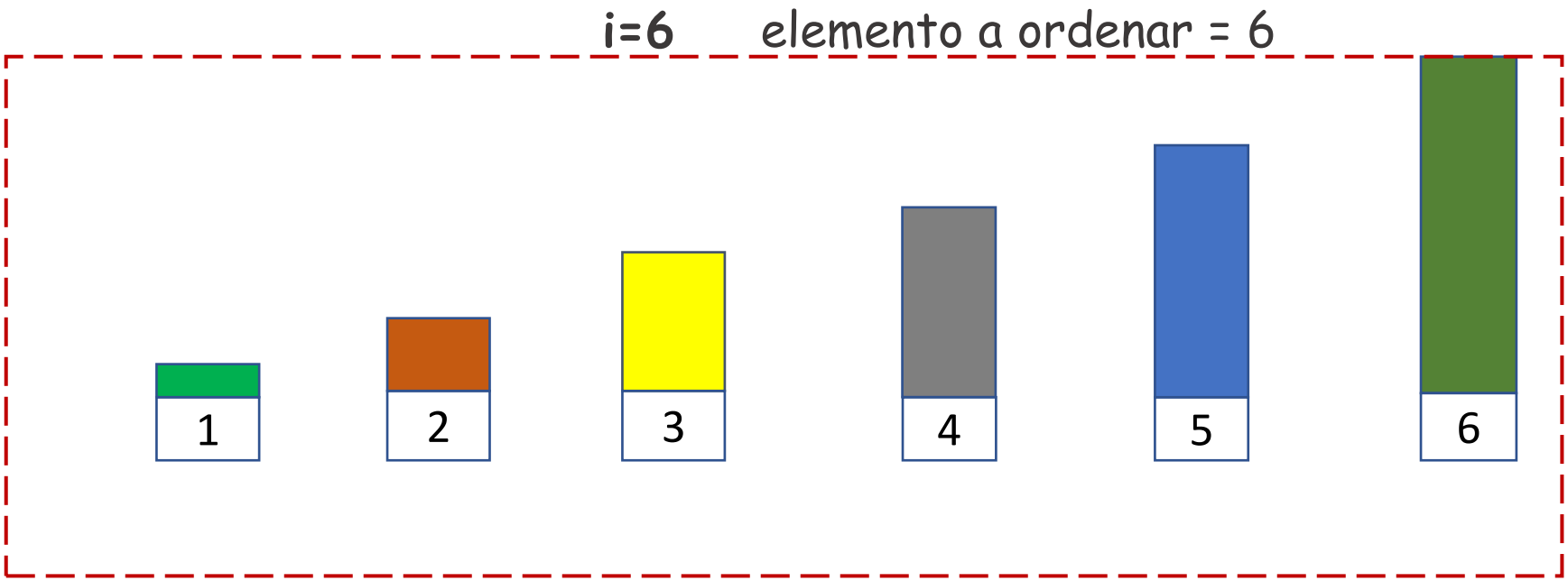
3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…



3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

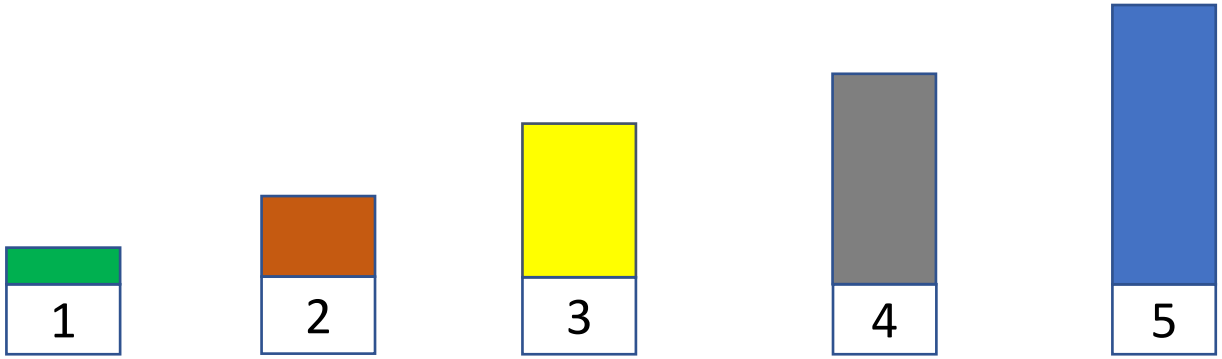
EJEMPLO: Ordenando de menor a mayor…



3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

EJEMPLO: Ordenando de menor a mayor…

i=6 elemento a ordenar = 6



pos final= 6

3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

3.1 PSEUCODIGO: Ordenar vector de n elementos de menor a mayor

```
Repetir desde i:=2 hasta n
   guardar elemento (a ordenar)
   j:= i-1
   Mientras (j > 0) y (v[j] > elemento a ordenar)
      v[j+1]:= v[j]
      j:= j - 1
   guardar elemento en v[j+1]
```

3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

3.2 Análisis de casos

Link al simulador de ordenación por inserción

http://lwh.free.fr/pages/algo/tri/tri_insertion_es.html

```
Repetir desde i:=2 hasta n
   guardar elemento (a ordenar)
   j:= i-1
   Mientras (j > 0) y (v[j] > elemento a ordenar)
      v[j+1]:= v[j]
      j:= j - 1
   guardar elemento en v[j+1]
```

¿Qué pasa si los datos están ordenados de menor a mayor ?

1	2	3	4	5	6

¿Qué pasa si los datos están ordenados de mayor a menor ?

6	5	4	3	2	1

¿Qué modificaciones haría en este algoritmo si quisiera ordenar un vector de registros?

Ejercitación

ACTIVIDAD 3:

a. Implementar el módulo **OrdenacionPorInsercion** utilizando el pseudocódigo analizado.

b. En el ProgramaVector.pas (creado en la clase 0)

   1. Incorporar el módulo **OrdenaciónPorInserción**.
   2. Invocar al módulo **OrdenacionPorInsercion**
   3. Mostrar el vector ordenado

Enviar a través de la Mensajería de Ideas, el archivo ProgramaVector.pas al docente asignado al grupo.

3. MÉTODO DE ORDENACIÓN POR INSERCIÓN

3.3 Análisis teórico

```
Repetir desde i:=2 hasta n
    guardar elemento (a ordenar)
    j:= i-1
    Mientras (j > 0) y (v[j] > elemento a ordenar)
        v[j+1]:= v[j]
        j:= j - 1
    guardar elemento en v[j+1]
```

Supongamos que **C**: Nro comparaciones y **M**: Nro de intercambios

Mejor caso

1	2	3	4	5	6

$n-1 <= C <= n(n-1)/2$

$2(n-1) <= M <= 2(n-1) + n(n-1)/2$

Peor caso

6	5	4	3	2	1

Ejercitación

ACTIVIDAD 4

Implementar un programa que procese la información de los participantes de un concurso de preguntas y respuestas (como máximo 20). De cada participante se lee el código de participante y su edad. El ingreso de los participantes finaliza cuando se lee el código -1.

a. Almacenar la información que se lee en un vector.
b. Mostrar la información almacenada.
c. Ordenar el vector de participantes por edad.
d. Mostrar el vector ordenado.
e. Eliminar del vector ordenado los participantes con edades entre 20 y 22. La solución debe ser eficiente con respecto al tiempo de ejecución.
f. Mostrar el vector resultante