ESTRUCTURAS DE DATOS

NOTAS SOBRE C++

STL: Iteradores

Manuel Montenegro Montes
Departamento de Sistemas Informáticos y Computación
Facultad de Informática – Universidad Complutense de Madrid

Tipos de iteradores

- Iteradores de entrada.
- Iteradores de salida.
- Iteradores hacia delante.
- Iteradores bidireccionales.
- Iteradores de acceso aleatorio.

Iteradores de entrada

Entrada

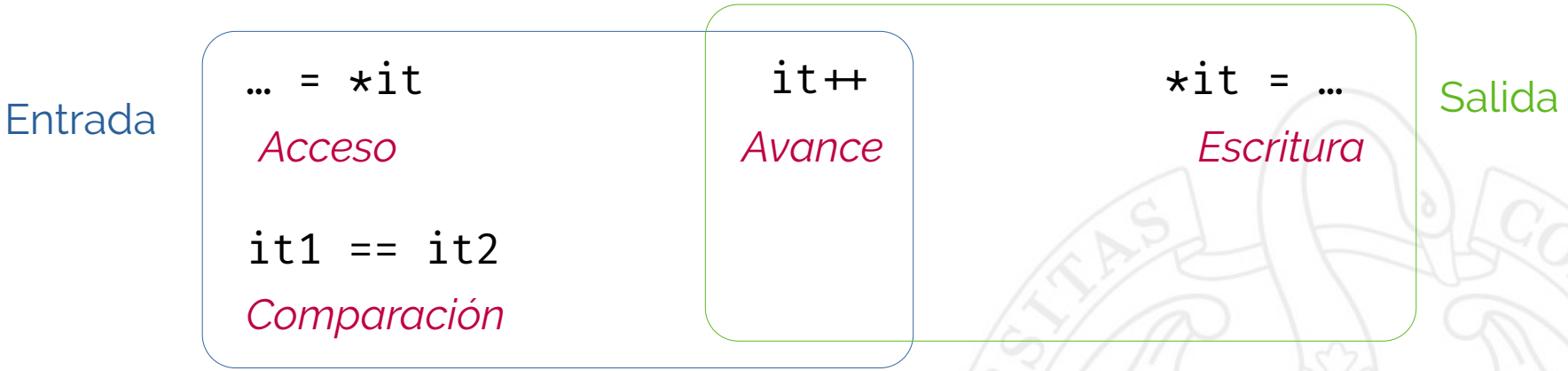
```
… = *it                    it++
```

Acceso *Avance*

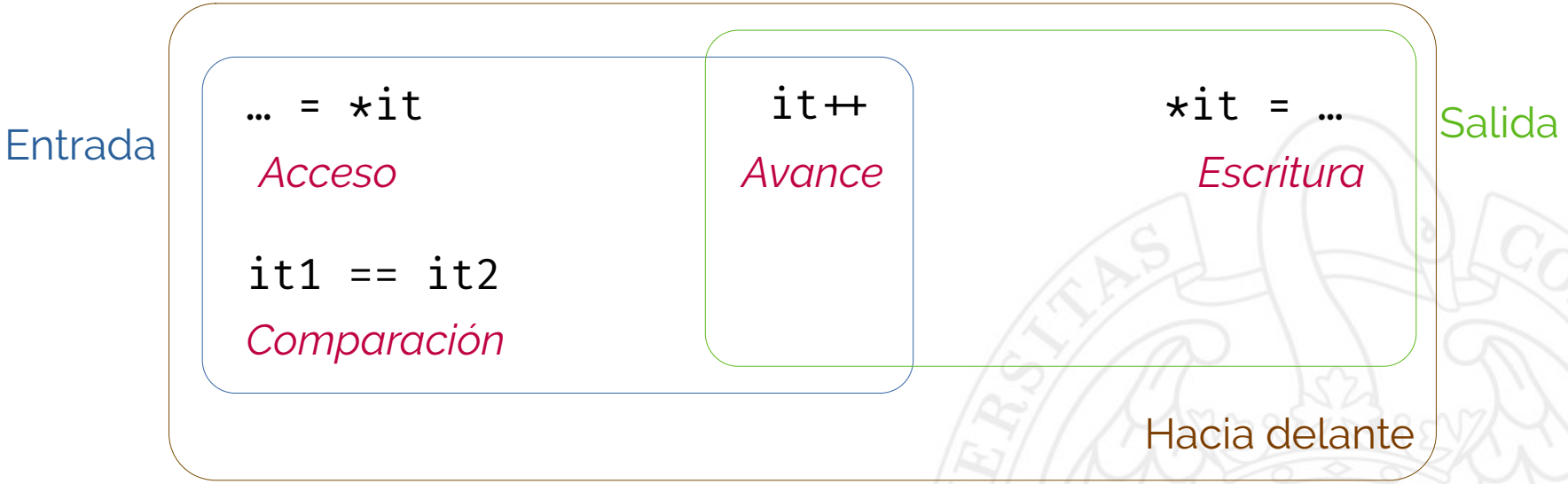
```
it1 == it2
```

Comparación

Iteradores de salida



Iteradores hacia delante



Iteradores bidireccionales

`… = *it`
Acceso

`it++`
Avance

`*it = …`
Escritura

`it1 == it2`
Comparación

Hacia delante

`it--`
Retroceso

Bidireccionales

Iteradores de acceso aleatorio

`… = *it` *Acceso*

`it++` *Avance*

`*it = …` *Escritura*

`it1 == it2` *Comparación*

Hacia delante

`it--` *Retroceso*

Bidireccionales

`it = it ± n` *Avance/retroceso por saltos*

Acceso aleatorio

Tipos de iteradores

- Cada implementación de TAD soporta un tipo de iterador determinado.

Expresión	Tipo de iterador
`vector::begin()`	Acceso aleatorio
`list::begin()`	Bidireccional
`deque::begin()`	Acceso aleatorio
`forward_list::begin()`	Hacia delante
`ostream_iterator`	Salida
`istream_iterator`	Entrada
Punteros	Acceso aleatorio

Iterador de salida: `ostream_iterator`

- Es un iterador asociado a un flujo de salida (fichero, salida estándar, etc.)
- Cada vez que se modifica el valor apuntado por el iterador, se realiza una operación de salida.
- Cada vez que se incrementa el iterador, no se hace nada.
- Es útil para la función `copy()`

Ejemplo

Flujo de salida

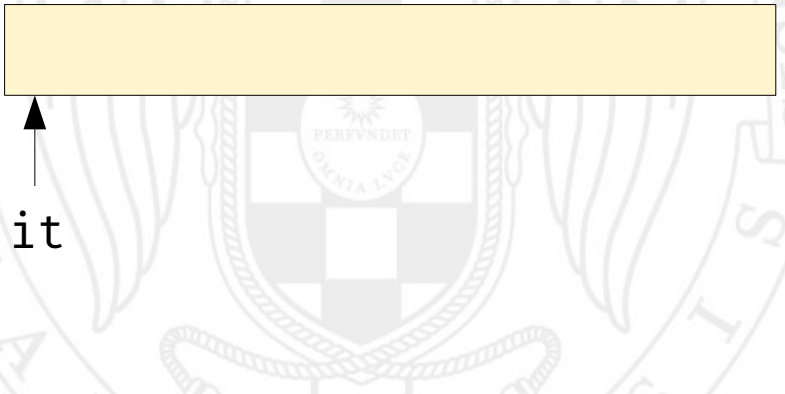
Separador

```
int main() {
  std::ostream_iterator<int> it(std::cout, " ");

  *it = 10;
  it++; // Opcional. No hace nada.
  *it = 20;
  it++; // Opcional. No hace nada.

  std::cout << std::endl;

  return 0;
}
```



Ejemplo

Flujo de salida

Separador

```
int main() {
  std::ostream_iterator<int> it(std::cout, " ");

  *it = 10;
  it++; // Opcional. No hace nada.
  *it = 20;
  it++; // Opcional. No hace nada.

  std::cout << std::endl;

  return 0;
}
```

Ejemplo

Flujo de salida

Separador

```
int main() {
  std::ostream_iterator<int> it(std::cout, " ");

  *it = 10;
  it++; // Opcional. No hace nada.
  *it = 20;
  it++; // Opcional. No hace nada.

  std::cout << std::endl;

  return 0;
}
```