



**CS1103**

**Programación Orientada a Objetos 2**

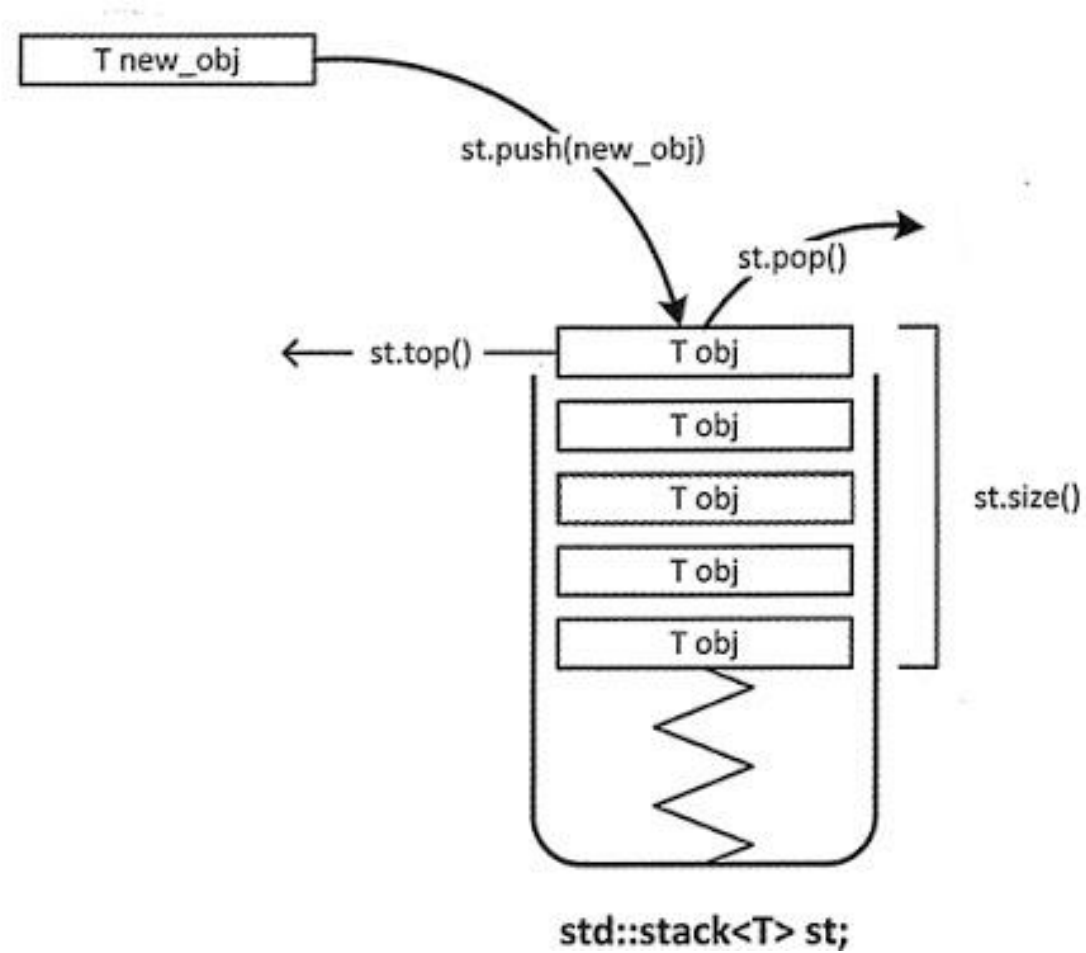
**Unidad 5: Semana 8**

**Librería Estándar - Adaptadores**

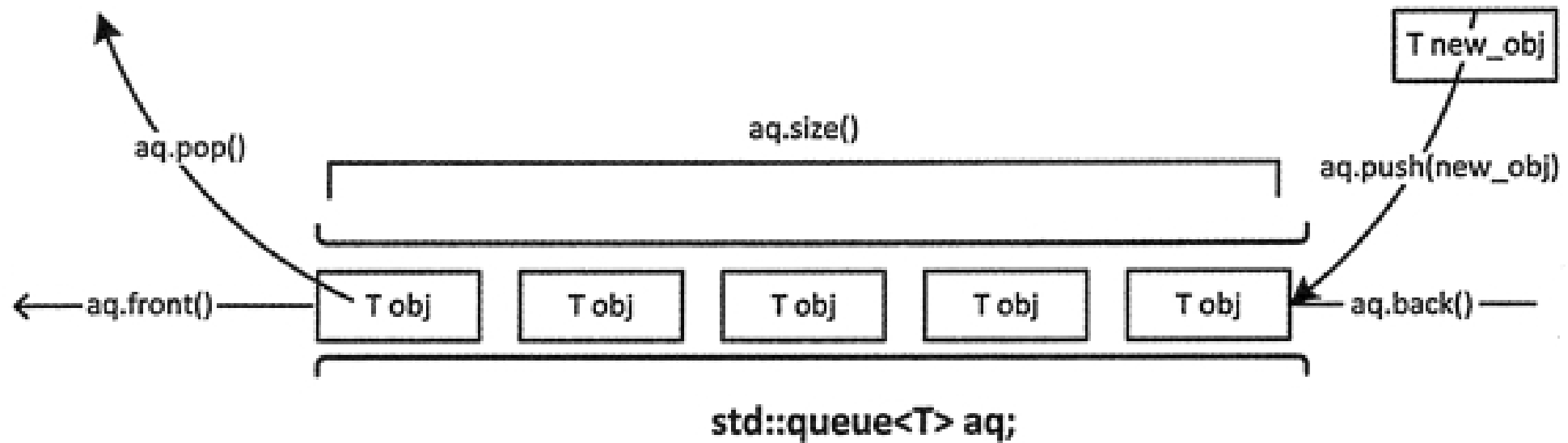
Rubén Rivas

---

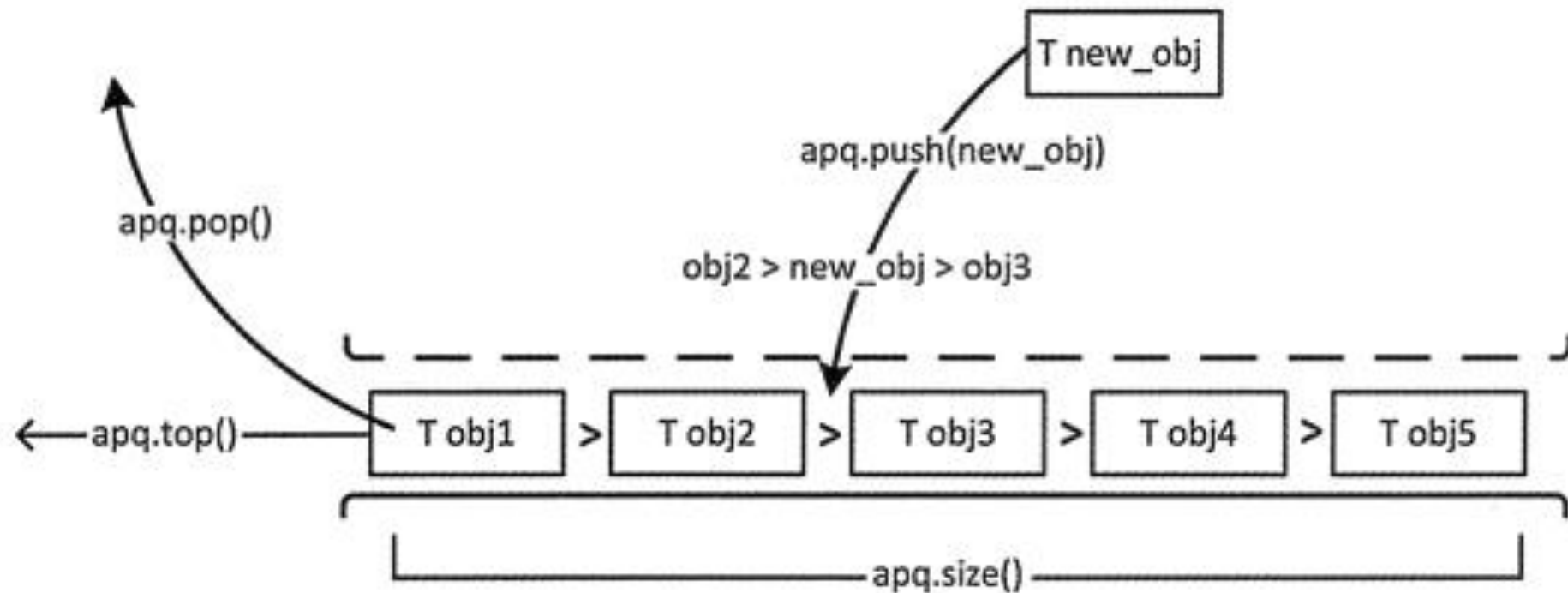
# std::stack



# std::queue



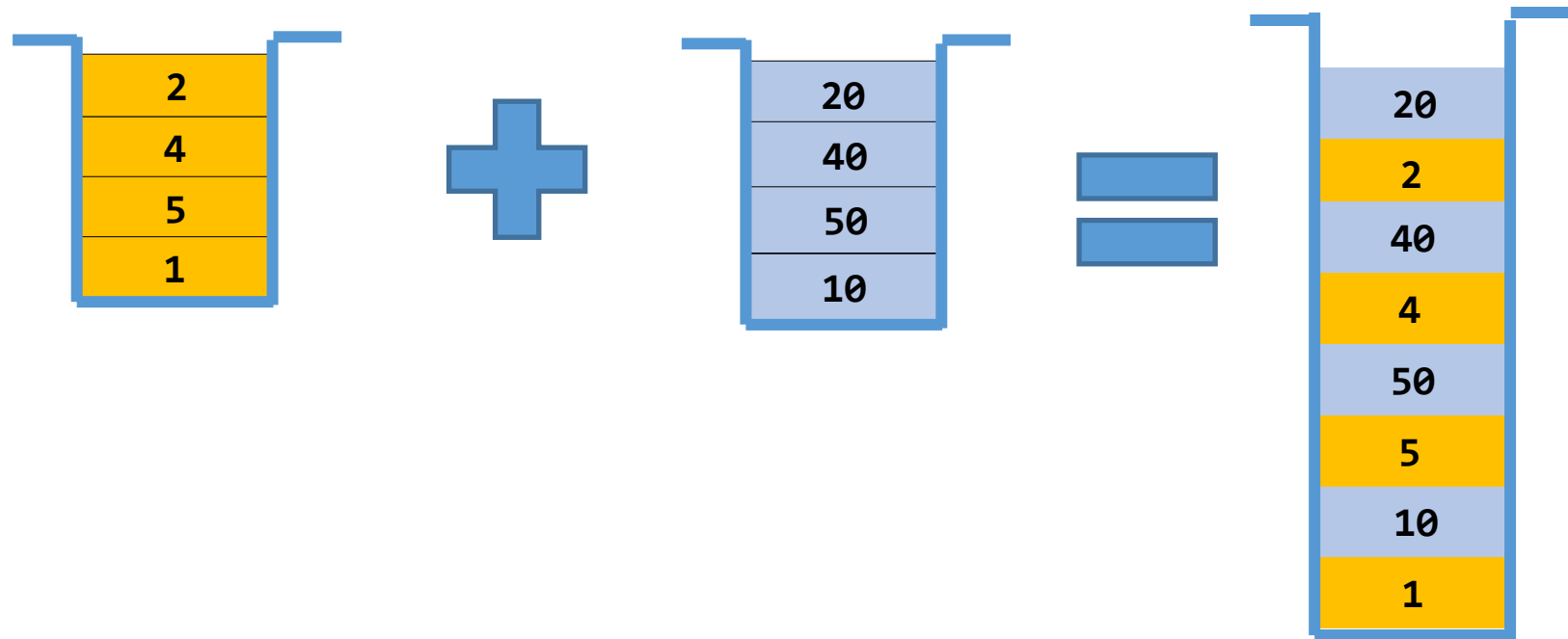
# std::priority\_queue



`std::priority_queue<T> apq;`

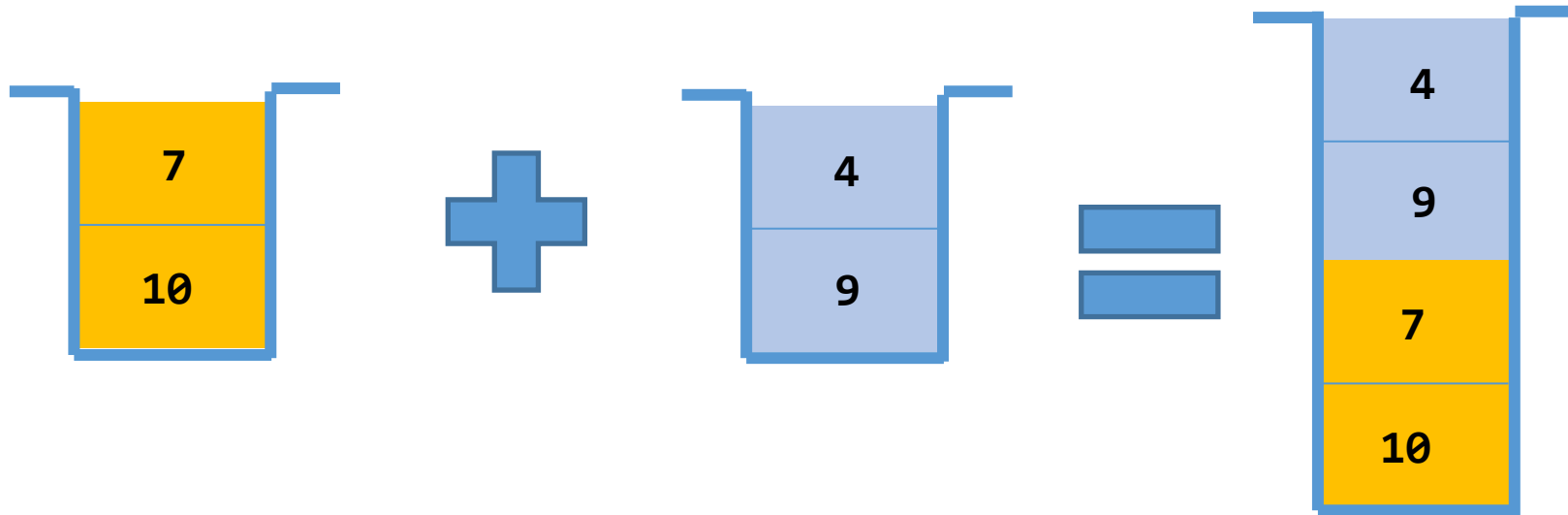
1. Elaborar un template de función **merge\_stacks**, que dada 2 stacks genere la unión con valores intercalados como muestra el diagrama.

```
stack<T> merge_stacks(stack<T> a, stack<T> b);
```



2. Elaborar un template de función **join\_stacks**, donde dada 2 stacks genere la unión como se muestra en el diagrama.

```
stack<T> join_stacks(stack<T> a, stack<T> b);
```



### 3. Elaborar una función que utilizando stack cumpla con las siguientes reglas:

Una letra significa un **push** y un \* significa un **pop**. Dada una secuencia de caracteres el programa deberá retornar la secuencia de caracteres extraídos por el **pop**.

Ejemplos:

EAS\*Y\*QUE\*\*\*ST\*\*\*IO\*N\*\*\* → SYEUQ TSAONIE

UN\*IV\*ERS\*IDADDEINGENIE\*RIAYTECN\*\*\*OLOGIA → NVSENCE

COMPUTER\*\*\*\*\*S\*\*CIEN\*\*CE → RETUPSMNE

```
string process_text_by_stack(string source);
```



## 4. Elaborar una función que utilizando queue cumpla con las siguientes reglas:

Una letra significa un **push** y un \* significa un **pop**. Dada una secuencia de caracteres el programa deberá retornar la secuencia de caracteres extraídos por el **pop**.

Ejemplos:

EAS\*Y\*QUE\*\*\*ST\*\*\*IO\*N\*\*\* → EASYQUESTION

UN\*IV\*ERS\*IDAD\*\* → UNIVE

COMPUTER\*\*\*\*\*S\*\*CIEN\*\*CE → COMPUTERS

```
string process_text_by_queue(string source);
```

## 5. Dada una ecuación, realizar una función basada en stack que determine si el balance de separadores es correcto.

Ejemplo:

$((10 + 20) + 30)$	Balanceado
$((10 +(30 * 1 + 2))$	No Balanceado
$(10 + (15+ 4/ 2 - 5)))$	No Balanceado
$\{ (6 + 20) + 4)$	No Balanceado
$\{ [ (4 - 5) * 3] + 14\}$	Balanceado

```
bool is_equation_balanced(string source);
```

## 6. Pila

Utilizando la clase **alumno\_t** que incluya los atributos **nombre**, **apellido** y **nota (int)**, se genere interativamente un deque.

Elaborar un template de función **find\_in\_stack** que permita ubicar un valor basado en cualquier criterio.

```
T find_in_stack(stack<T> s, function<bool(T)> cmpr);
```

```
// Crear un stack basado en un deque de alumnos  
auto a1 = find_in_stack(q1, [] (alumno_t a) {  
    a.apellido == "Perez"  
});
```



## 7. Elaborar una función basada en stack que tome un expresión sin el paréntesis izquierdo y que retorne la expresión infijo con los paréntesis completos.

- Ejemplo:

$10 + 20) * 5 + 3) \rightarrow (10 + 20) * (5 + 3)$

$10 + 30) * 30) \rightarrow (10 + 30) * (30)$

$10 + 15) * 2 - 5)) \rightarrow ((10 + 15) * (2 - 5))$

```
string complete_equation(string source);
```

