

Clase #25 de 27

Módulos & Undo

Nov 1ro, Jueves



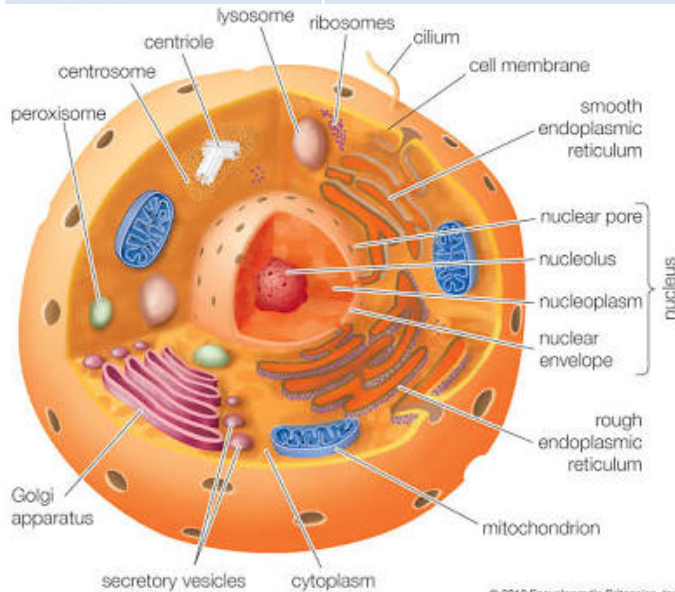
Agenda para esta clase

- Módulos
- Caso de Estudio: Undo de un Cli

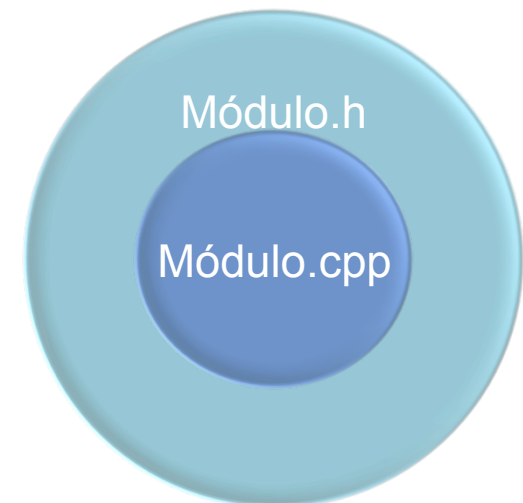
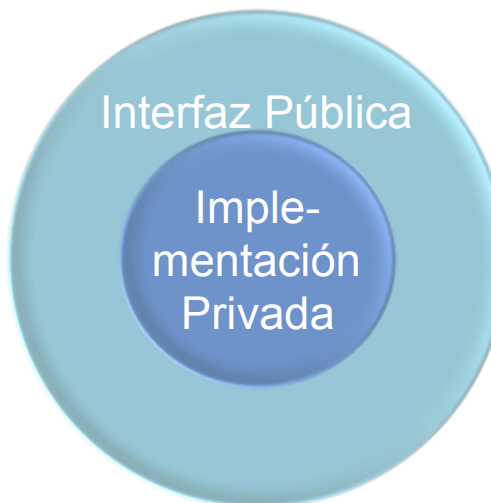
Módulos

Módulos en C++ y Analogía con Células

| Sistema | Célula | Módulo | C++ |
|--|--|--|---|
| <ul style="list-style-type: none"> Sistemas abiertos Interfaces Parte pública Parte privada Relaciones con otros sistemas | <ul style="list-style-type: none"> Membrana celular Pasaje de información a través proteínas de transporte en Núcleo con estructura interna privada | <ul style="list-style-type: none"> Interfaz, contrato, Invocaciones Envío de mensajes Estado Comportamiento Elementos privados | <ul style="list-style-type: none"> .h (Membrana) <ul style="list-style-type: none"> Tipos públicos Prototipos públicos Obejtos públicos .cpp (Núcleo) <ul style="list-style-type: none"> Estructura de datos privada Método (i.e., forma de responder el mensaje) privado Objetos, Tipos y Funciones auxiliares |



© 2010 Encyclopædia Britannica, Inc.



Dependencias entre Módulos

- Sistemas
 - Conjunto de elementos relacionados con un objetivo en común
 - Jerarquía: Sistemas y Subssistemas
- Programa
 - Conjunto de módulos relacionados entre sí con un objetivo en común
 - Jerarquía
- C++
 - Relación denotata por includes
 - Include que importa
 - "Depende de"
 - Include que exporta
 - "Implementa"

Atributos de Calidad de Módulos

- Alta Cohesión
 - Funcional
- Bajo Acoplamiento
 - Pocas dependencias
 - Dependencia sobre abstracciones, no sobre implementaciones
- <https://thebojan.ninja/2015/04/08/high-cohesion-loose-coupling>
- Ocultamiento de Información
 - Que el cliente no necesite conocer los detalles de la implementación para usar la abstracción
- Encapsulamiento
 - Bloquear al cliente el acceso a los detalles de la implementación
 - Keyword *static* para cerrar los poros de la membrana
 - *Forward declaration* para punteros a objetos con estructura desconocida

Undo en un CLI

Command Line Interpreter

Caso de Estudio

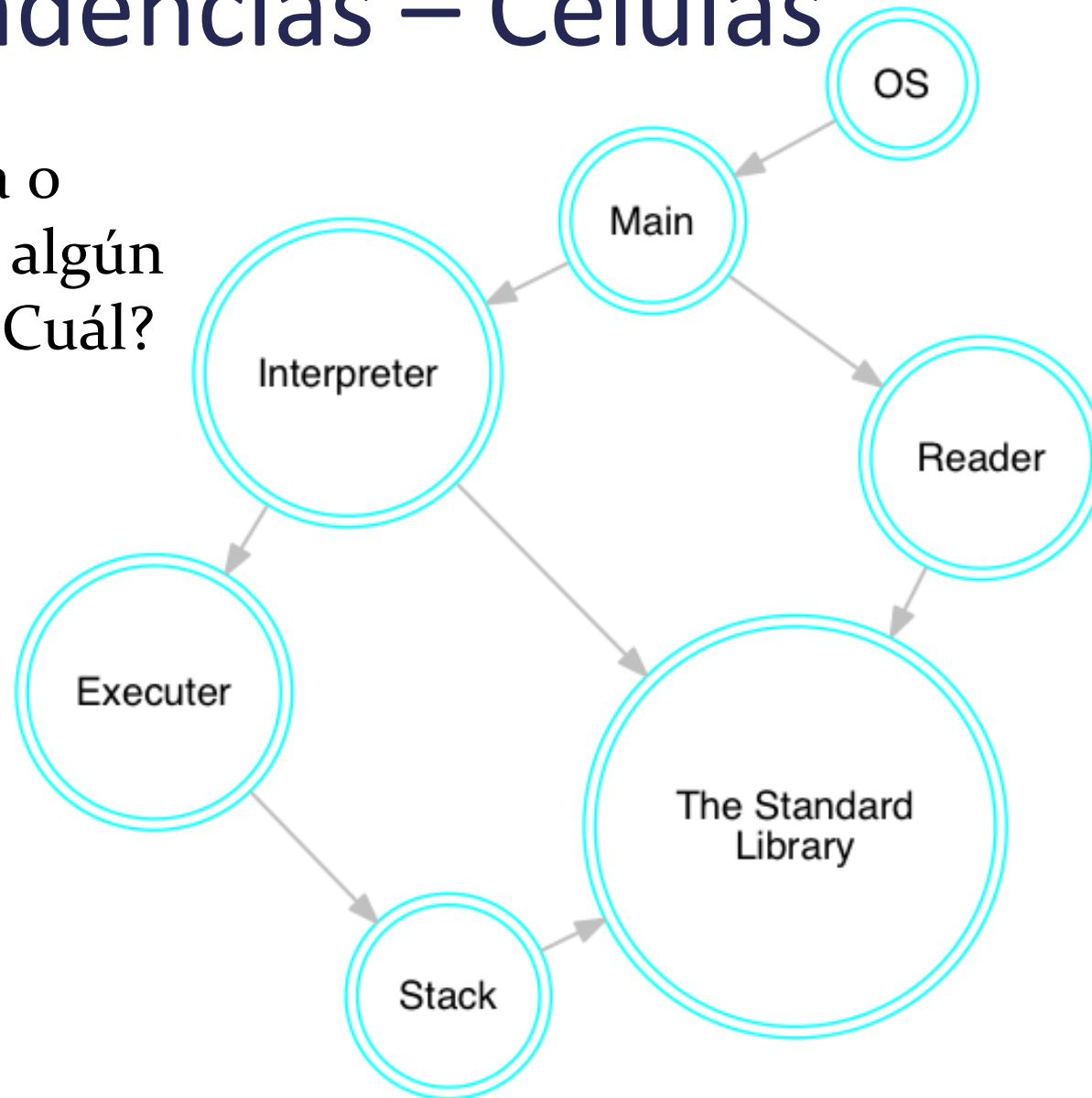
Cli – Uso y Diseño

```
$ cli
> comando1
comando1
> comando2
comando2
> comando3
comando3
> undo
comando3 undone
> comando4
comando4
> undo
comando4 undone
> undo
comando2 undone
> undo
comando1 undone
> undo
No hay nada para deshacer
> quit
Bye!
$
```

- Comandos
 - Echo
 - Undo
 - Quit
- Diseño
 - Diseñar Módulos
 - Diseñar Interfaces
 - Diseñar Relaciones
- Backlog de features
 - Comandos Case insensitive
 - Comando Redo
 - Comando Repeat / Do Over
 - Historial de comandos entre sesiones
 - Save
 - Load
 - Evaluación de expresión en notación polaca inversa
 - Evaluación de expresiones en notación infija
 - Variables

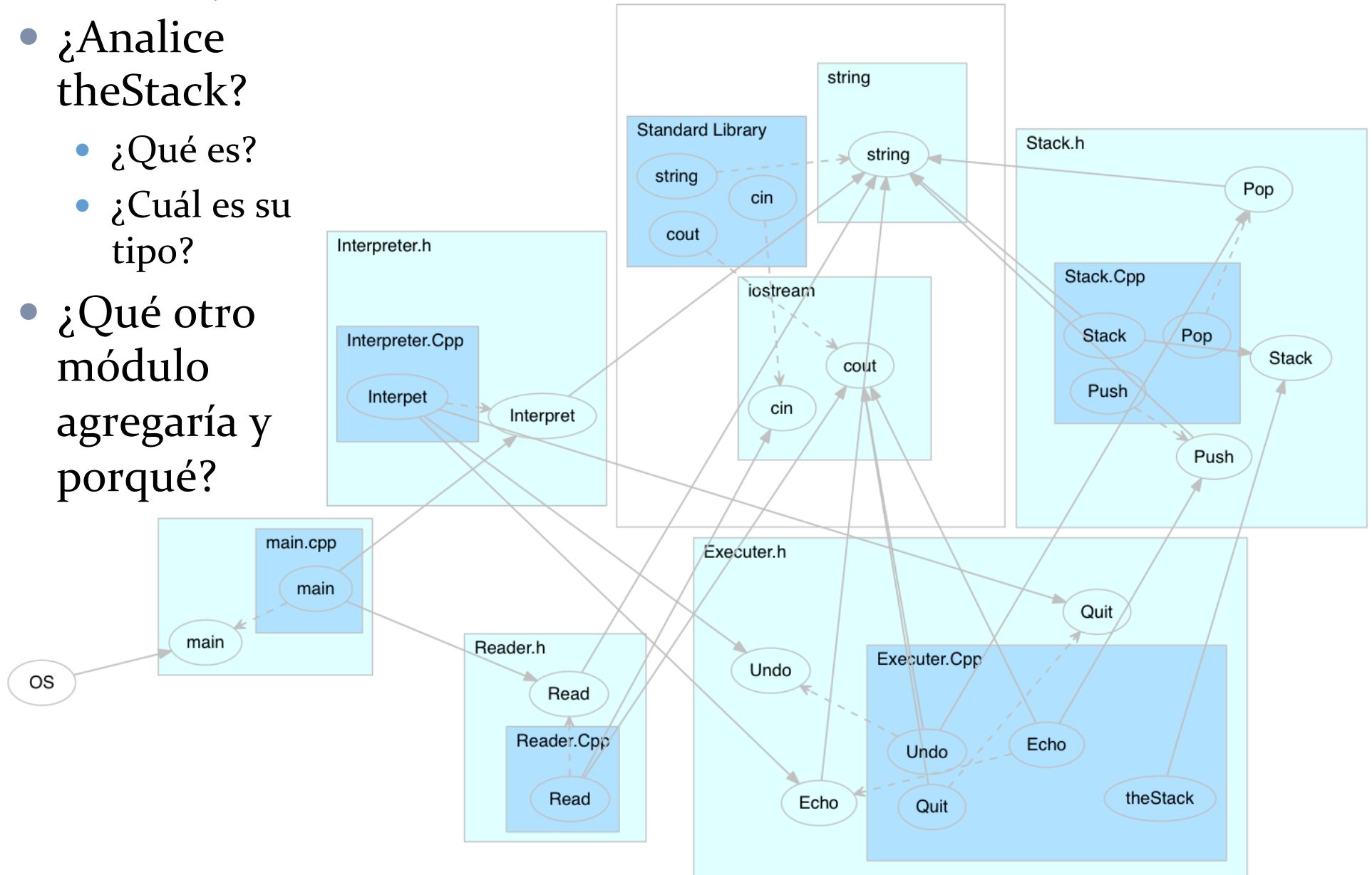
Sistema de Módulos y Dependencias – Células

- ¿Agregaría o removería algún módulo? ¿Cuál? ¿Porqué?

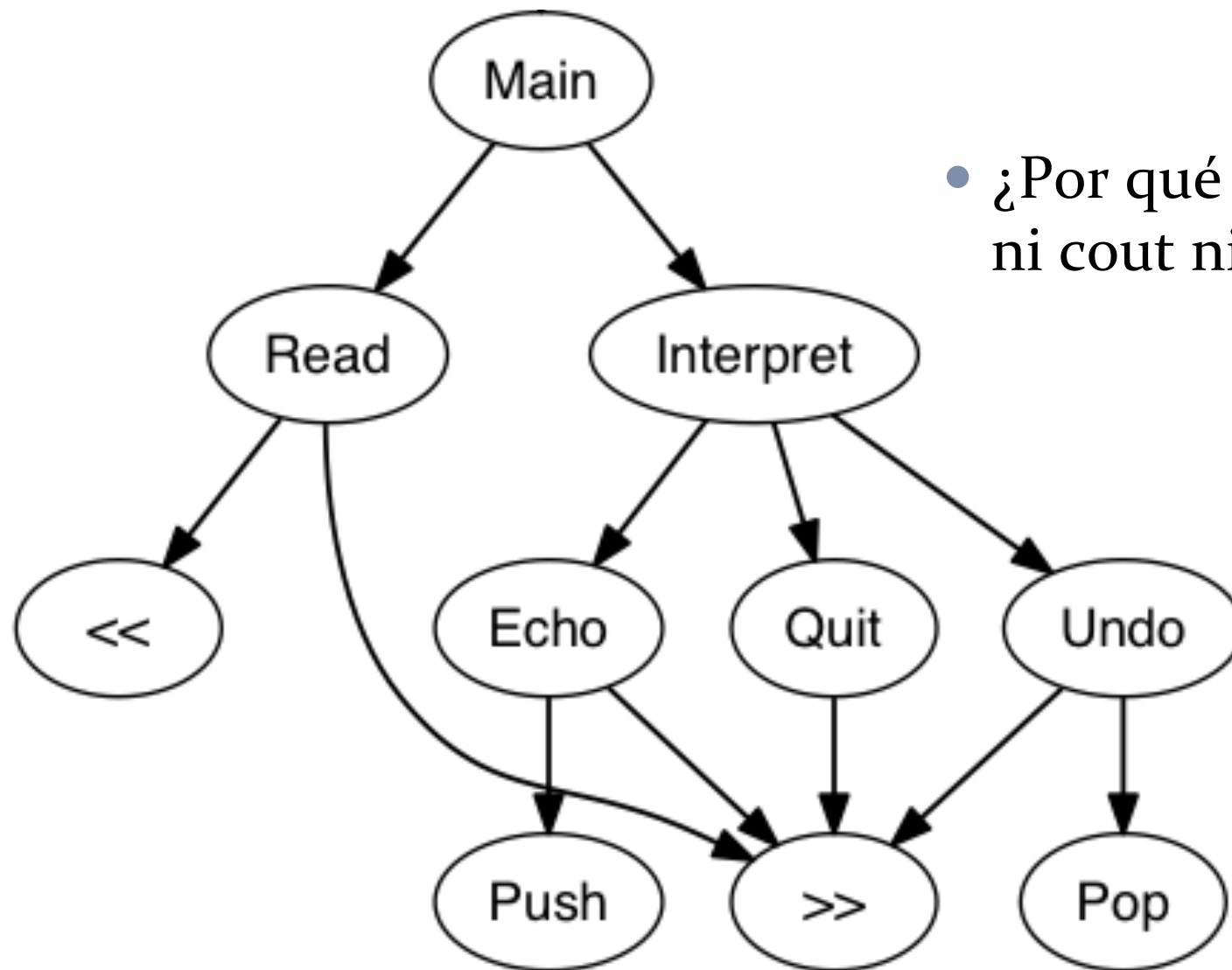


- ¿Analice theStack?
 - ¿Qué es?
 - ¿Cuál es su tipo?
- ¿Qué otro módulo agregaría y porqué?

- ¿Analice theStack?
 - ¿Qué es?
 - ¿Cuál es su tipo?
- ¿Qué otro módulo agregaría y porqué?



Structure Chart – Carta Estructurada



- ¿Por qué no están ni cout ni cin?



¿Consultas?



Fin de la clase