

Clase #26 de 27

Historial de Comandos & Flujo de Líneas

Noviembre 8, Jueves



Agenda para esta clase

- Caso de Estudio: Cli Redo – Dos Stacks y Persistencia
- Flujo de Líneas

CLI con Stacks de Undo y Redo, y Persistencia de Historial de Comandos

Caso de Estudio

Cli – Redo, Repeat, Historial de comandos persistente

```
$ cli
> comando1
comando1
> comando2
comando2
> comando3
comando3
> repeat
comando3
> undo
comando3 undone
> undo
comando3 undone
> undo
comando2 undone
> undo
comando1 undone
> redo
comando1
> redo
comando2
> redo
comando3
> redo
comando3
> redo
No hay nada para rehacer
> quit
Bye!
$ cli
> undo
comando3 undone
> undo
comando3 undone
> quit
Bye!
$ cli
> undo
comando2 undone
```

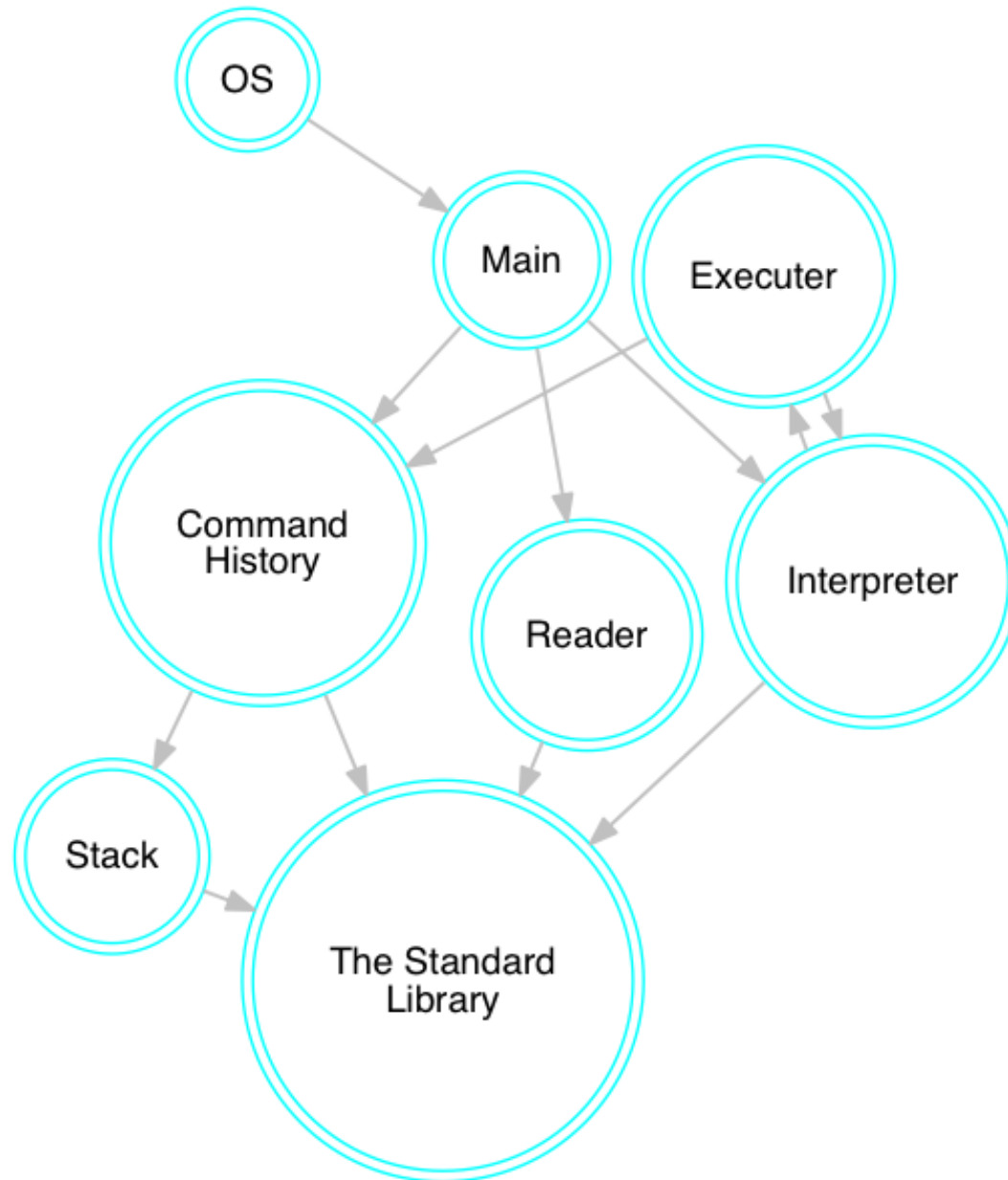
- Luego del primer repeat
 - Undo Stack
 - comando3
 - comando3
 - comando2
 - comando1
- Luego del primer redo
 - Undo Stack
 - comando1
 - Redo Stack
 - comando2
 - comando3
 - comando3
- Centinela de pila vacía
- El historial persiste entre sesiones
 - Quit
 - SaveCommandHistory
 - Archivos de líneas (renglones)
 - Líneas de longitud acotada
 - Líneas de longitud no acotada
 - Main
 - OpenCommandHistory.



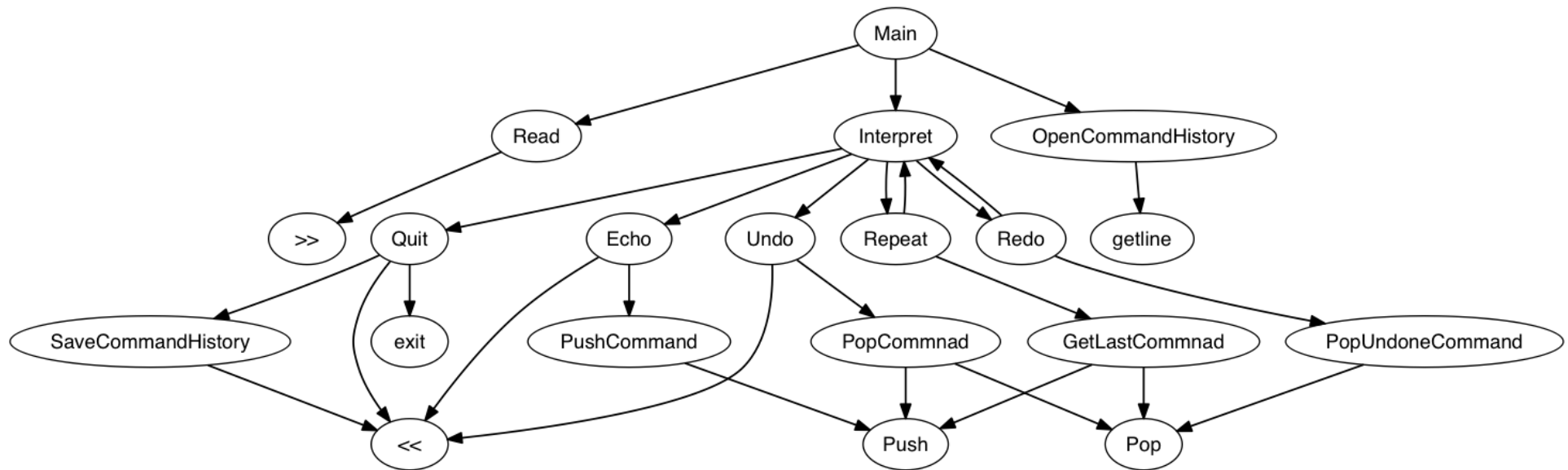
Historial de Comandos

- Quit lo guarda
 - SaveCommandHistory
- Main lo abre
 - OpenCommandHistory
- El responsable es el módulo CommandHistory, que conoce la estructura interna
- Precondiciones, Poscondicones, y Excepciones sobre disponiblidad de archivos.

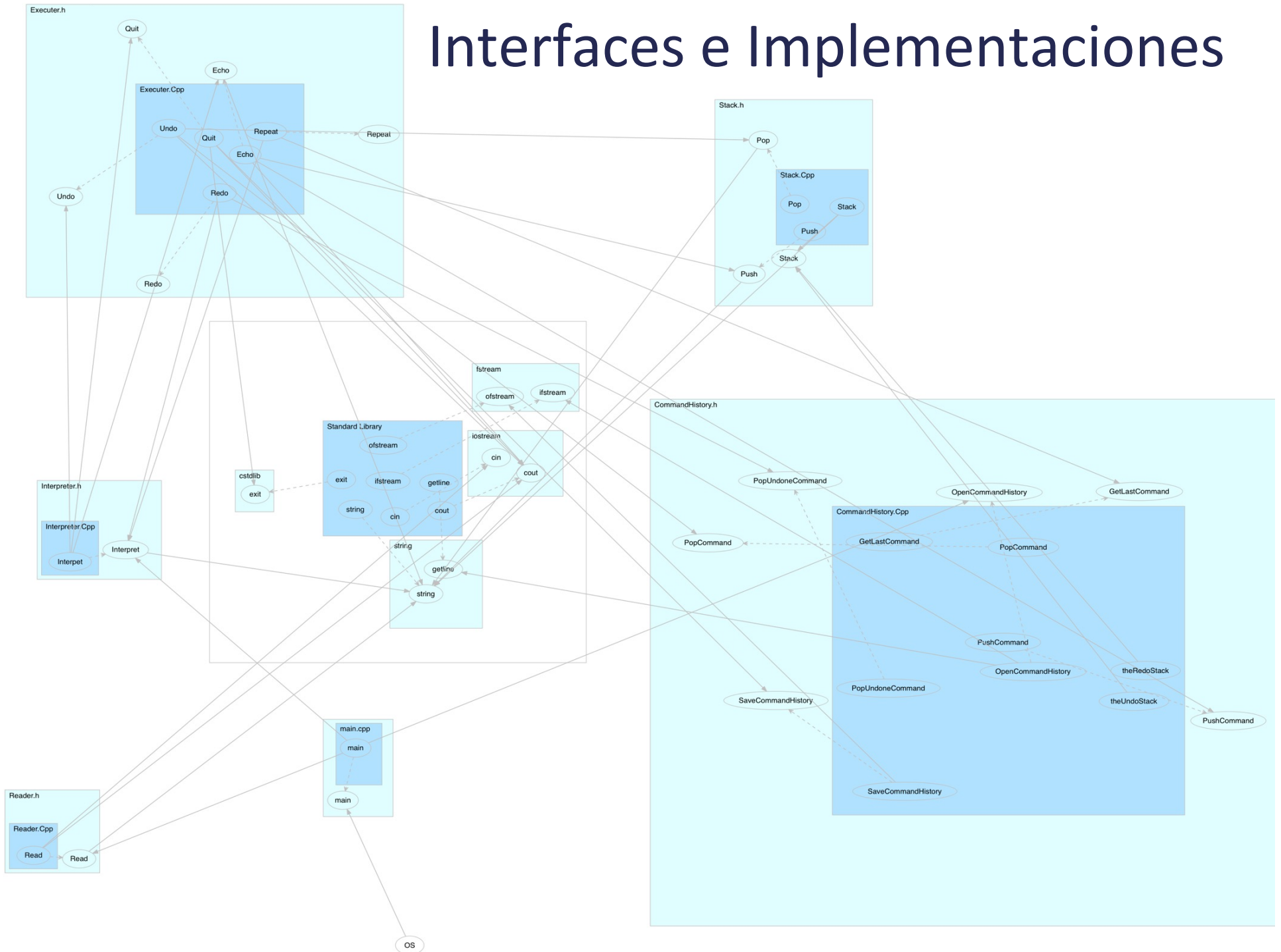
Sistema de Módulos y Dependencias – Células



Structure Chart – Carta Estructurada



Interfaces e Implementaciones



Flujo de Líneas

Getline

Copiar líneas usando Strings

```
#include <fstream>
#include <string>

int main(){
    std::ifstream in{"input.txt"};
    std::ofstream out{"output.txt"};

    for(std::string s; std::getline(in, s); )
        out << s << '\n';
}
```



¿Consultas?



Fin de la clase