

# Mediator Design Pattern

---

PATRON DE DISEÑO MEDIADOR



/luisdavaría

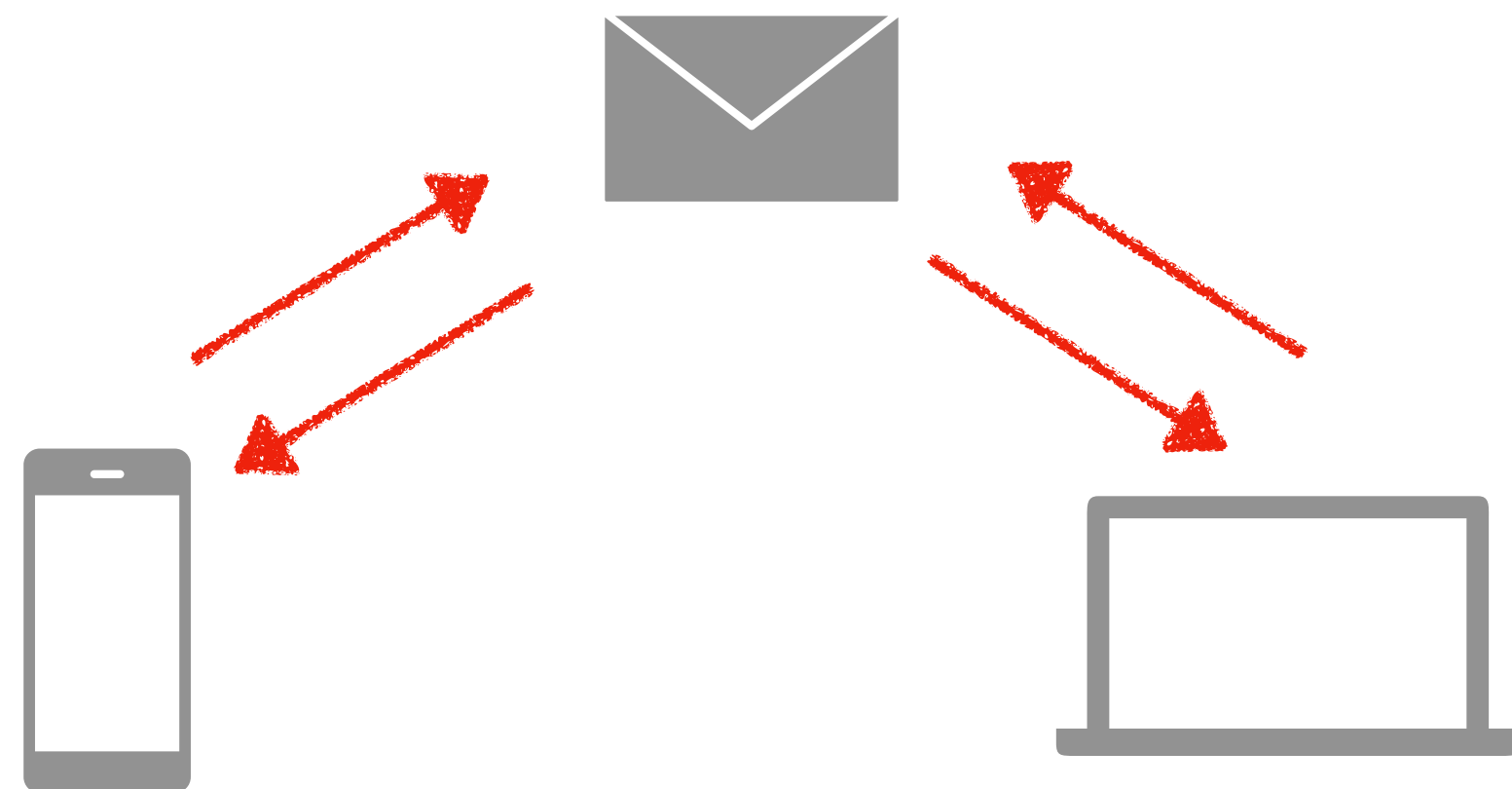


/Davaría



# Mediator

"Facilita la comunicación entre objetos"



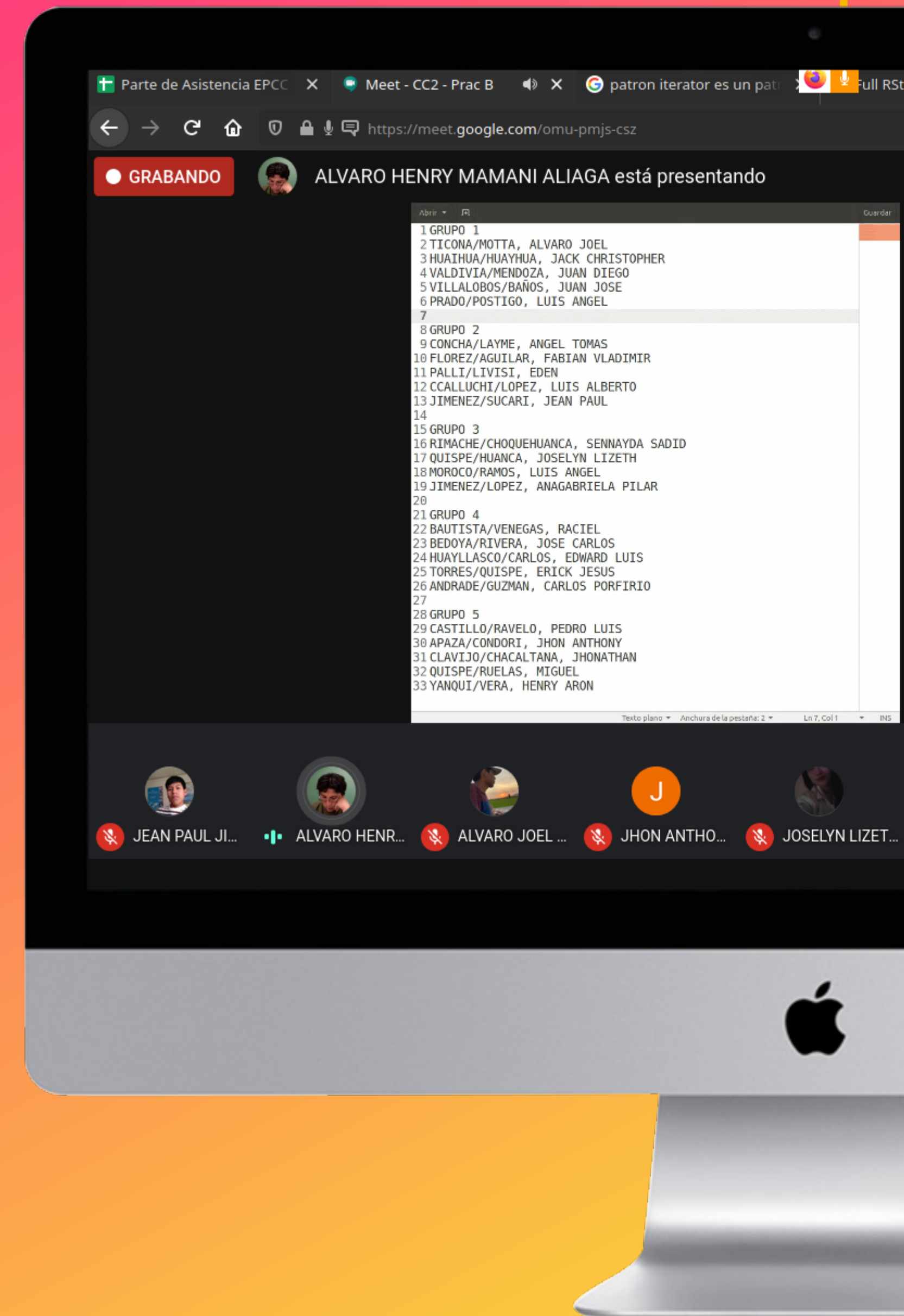
Objeto que encapsula cómo un conjunto de objetos interactúan



Variar interacción de forma independiente



Patron de diseño de tipo Comportamiento





# Programando Google Meet

Un ejemplo clásico y más adecuado del Patrón de diseño de Mediator.



1

Una sala de chat donde sus participantes pueden interactuar.



2


Los participantes pueden referirse entre si pero estas desaparecerán.



3

Participantes deben referir a un componente central que facilite la comunicación y sea el mediador.

# Clase Abstracta ~ ChatRoom

```
  
class ChatRoom {  
    public:  
        // Transmitir  
        virtual void broadcast(string from, string msg)= 0;  
};
```





# Clase Person ~ Participantes

```
class Person : public ChatRoom{
    public:
        string          name;
        ChatRoom*       room{nullptr};

        void broadcast(string , string ){};
        Person(string _name) { this->name = _name;}
        void say(string msg) { room->broadcast(name, msg);}
        void receive(string from, string msg) {
            string s{from + ": \"" + msg + "\""};
            cout << "[Chat de " << name << "]" << s << "\n";
        }
};
```



# Clase GoogleMeet

## Mediator

- Objeto que encapsula cómo un conjunto de objetos interactúan
- Variar interacción de forma independiente

```
class GoogleMeet : public ChatRoom
{
    public:
    vector<Person*>    people;
    // Transmitir
    void broadcast(string from, string msg) {
        for (auto p : people)
            if (p->name != from)
                p->receive(from, msg);
    }
    // Unen al chat
    void join(Person *p) {
        string join_msg = p->name + " se unio al chat";
        broadcast("room", join_msg);
        p->room = this;
        people.push_back(p);
    }
};
```





# Google Meet

```
int main() {
    GoogleMeet room;

    Person john("John");
    Person jane("Jane");

    room.join(&john);
    room.join(&jane);
    john.say("Jane hiciste la tarea?");
    jane.say("oh, hey john");

    Person simon("Simon");
    room.join(&simon);
    simon.say("Hola a todos!");

    return 0;
}
```



/Davaria/mediator

# Ventajas de Mediator

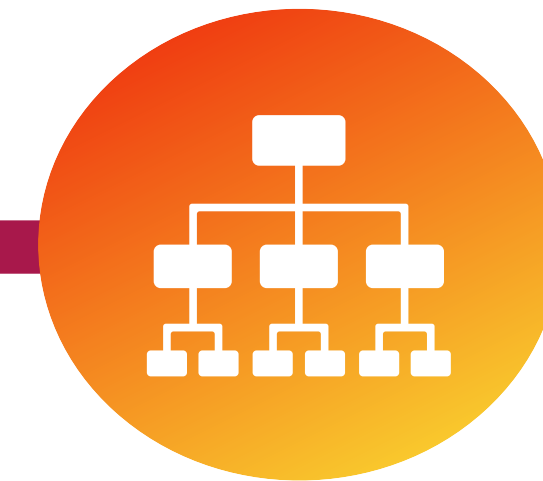


Supera

- Observer funciona en un relación uno a mucho.
- Mediador muchos a muchos.



Reduce la complejidad de la comunicación entre los diferentes componentes de un sistema.



Puede reemplazar cualquier componente en el sistema sin afectar a otros componentes y sistema.



Fondo de color



Tipografías

**TITULOS /Thonburi**

Texto/Menlo