

Centro de Datos

CLIPS

Tomás Rial Costa 35602105Z tomasrialacademica@gmail.com
Adán González Rodríguez 44658877F adan.rozaguez@gmail.com



ÍNDICE

1. Resumen del problema
2. Alcance del Sistema
3. Diseño
 - 3.1. Centro de Datos
 - 3.2. Templates
 - 3.3. Reglas
 - 3.4. Hechos Iniciales
4. Funcionamiento
5. Manual de Uso
 - 5.1. Requisitos del programa
 - 5.2. Uso del programa
6. Conclusiones
7. Bibliografía
8. Problemas encontrados en el desarrollo

Resumen del Problema

El sistema de control inteligente simula un entorno de centro de datos con distintas zonas y elementos críticos (como racks de servidores, módulos de aire, y zonas sensibles), asegurando condiciones óptimas de temperatura, iluminación y acceso, así como la capacidad de responder a situaciones de emergencia (incendios) usando la librería clipspy.

Alcance del Sistema

Control de Accesos:

- Debemos asegurarnos de que para acceder a una zona en concreto el nivel del usuario que quiere acceder a esta zona sea igual o superior al nivel de acceso de la zona.
- También debemos tener constancia de cómo fluctúa la ocupación de las zonas en función de los movimientos de los distintos usuarios por el Centro de Datos.

Control de Climatización:

- Los incrementos de temperatura deben realizarse de la siguiente forma ($\pm 1^{\circ}\text{C}$) y únicamente cuando los valores de temperatura están fuera de los límites establecidos. (Menor de 20°C y Mayor que 25°C se encargará de ajustar la temperatura hasta 22°C)

Detección de Incendios:

- Cuando se detecta un incendio se debe evacuar la zona.
- Una vez no haya nadie en la zona (Ocupación = 0) se solucionará el desastre usando Gas cuando la zona sea sensible y/o usando agua cuando la zona sea no sensible (Normal).

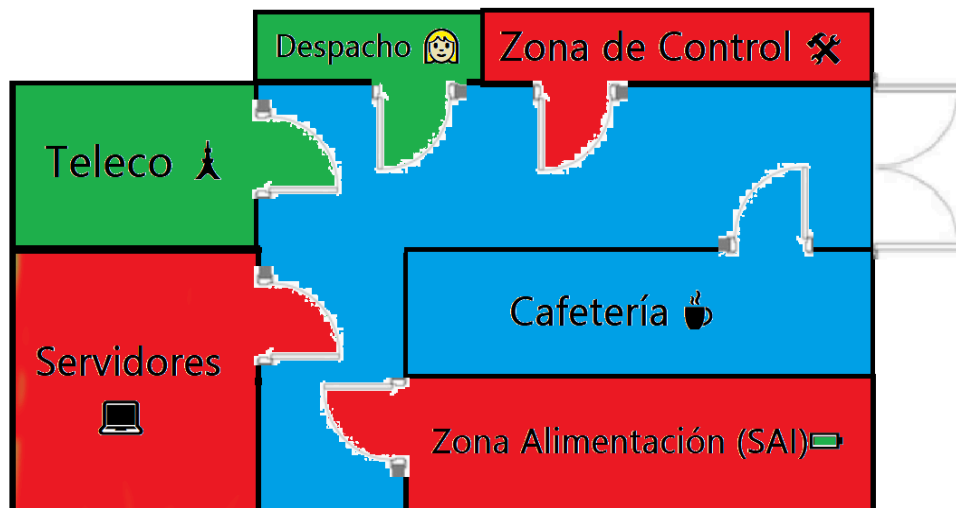
Control de Alimentación:

- Debemos controlar que siempre que haya al menos 1 ocupante en una zona la iluminación de esa zona esté encendida.
- En caso de que la ocupación de una sala sea 0 la luz debe estar apagada.
- Debemos reajustar los valores del voltaje de los Racks que tengan un voltaje anómalo (Si sus valores salen del rango 210-230 V).

Diseño

Centro de Datos

El centro de datos que vamos a simular tiene la siguiente forma:



Donde podemos observar las siguientes zonas:

- Pasillo y Cafetería (Azules == Nivel de Acceso Necesario = 1)
- Telecomunicaciones y Despacho (Verdes == Nivel de Acceso Necesario = 2)
- Servidores, Zona de Control y SAI (Rojas == Nivel de Acceso Necesario = 3)

Templates

En el sistema se definen los siguientes templates:

1. Zona:

- Nombre identificador de la zona
- Nivel de acceso a dicha zona (1-3)
- Temperatura en esa sala
- Ocupación total de esa zona en ese momento
- Tipo de sala (normal / sensible)
- Estado de desastre actual (normal / activo)
- Iluminación de esa sala (apagada/encendida)

2. Usuario:

- a. Nombre del usuario
- b. Nivel de acceso a zonas

3. Modulo-Aire:

- a. Sala en la que se encuentra
- b. Temperatura Objetivo
- c. Estado (encendido/apagado)

4. Rack:

- a. Id del Rack
- b. Voltaje

5. Intento-Acceso:

- a. Usuario que intenta acceder
- b. Zona a la que intenta acceder

Reglas

Las reglas que diseñamos para los diferentes controles del sistema son las siguientes:

- Reglas para verificar el acceso:

```
; Regla para verificar acceso denegado
(defrule acceso-denegado
  ?intento <- (intento-acceso (usuario ?nombre-usuario) (zona-nueva ?nombre-zona-nueva) (zona-antigua ?nombre-zona-antigua))
  ?usuario <- (usuario (nombre ?nombre-usuario) (nivel-acceso ?nivel-acceso))
  ?zona-nueva <- (zona (nombre ?nombre-zona-nueva) (nivel ?nivel-zona-nueva))
  (test (< ?nivel-acceso ?nivel-zona-nueva))
  =>
  (printout t "Acceso denegado para " ?nombre-usuario " a la zona " ?nombre-zona-nueva ". Nivel de acceso insuficiente." crlf)
  (retract ?intento)
)

; Regla para verificar acceso permitido
(defrule acceso-permitido
  ?intento <- (intento-acceso (usuario ?nombre-usuario) (zona-nueva ?nombre-zona-nueva) (zona-antigua ?nombre-zona-antigua))
  ?usuario <- (usuario (nombre ?nombre-usuario) (nivel-acceso ?nivel-acceso))
  ?zona-nueva <- (zona (nombre ?nombre-zona-nueva) (ocupacion ?ocupacion-nueva) (nivel ?nivel-zona-nueva))
  ?zona-antigua <- (zona (nombre ?nombre-zona-antigua) (ocupacion ?ocupacion-antigua))
  (test (>= ?nivel-acceso ?nivel-zona-nueva))
  =>
  (printout t "Acceso permitido para " ?nombre-usuario " a la zona " ?nombre-zona-nueva ". Nivel de acceso suficiente." crlf)
  (modify ?zona-nueva (ocupacion (+ ?ocupacion-nueva 1)))
  (modify ?zona-antigua (ocupacion (- ?ocupacion-antigua 1)))
  (retract ?intento)
)
```

Estas dos reglas se encargan de controlar que a las salas solo puedan acceder usuarios con nivel de acceso igual o superior al de la sala. Se tiene en cuenta la sala a la que va y de la que viene para poder mantener la ocupación de salas actualizada. Cuando el nivel de acceso es insuficiente simplemente se muestra un mensaje. En cambio cuando es suficiente, además del mensaje, se procede a la actualización de ocupaciones de las salas. Se usa retract para que no se den bucles infinitos cuando se accede a la sala.

- Reglas para control de climatización:

```
; Control de climatizacion

(defrule control-temperatura
  ?zona <- (zona (nombre ?nombre) (temperatura ?temp))
  ?modulo <- (modulo-aire (sala ?nombre) (temperatura-objetivo 22) (estado "apagado"))
  (or (test (< ?temp 20)) (test (> ?temp 25)))
  =>
  (modify ?modulo (estado "encendido"))
  (printout t crlf "Temperatura en " ?nombre " no ideal, aire encendido" crlf))

(defrule ajustar-temperatura
  ?modulo <- (modulo-aire (sala ?sala) (temperatura-objetivo 22) (estado "encendido"))
  ?zona <- (zona (nombre ?sala) (temperatura ?temp))
  (test (!= ?temp 22))
  =>
  (if (< ?temp 22)
    then (progn
      (modify ?zona (temperatura (+ ?temp 1)))
      (printout t "Ajustando temperatura en " ?sala " de " ?temp " a " (+ ?temp 1) crlf))
    else (progn
      (modify ?zona (temperatura (- ?temp 1)))
      (printout t "Ajustando temperatura en " ?sala " de " ?temp " a " (- ?temp 1) crlf))))

(defrule temperatura-ajustada
  ?modulo <- (modulo-aire (sala ?sala) (temperatura-objetivo 22) (estado "encendido"))
  ?zona <- (zona (nombre ?sala) (temperatura ?temp))
  (test (= ?temp 22))
  =>
  (modify ?modulo (estado "apagado"))
  (printout t "Temperatura en " ?sala " ideal, aire apagado " crlf))
```

Con las tres reglas mostradas conseguimos manejar el aire acondicionado, lo cual nos permite controlar la temperatura de las zonas. Cuando la temperatura sale del rango deseado se enciende el módulo de aire.

Con esto se dispara otra regla, la cual va subiendo la temperatura hasta llegar a la ideal. Cuando llega a la ideal, para evitar bucles, hicimos otra regla que se dispara y apaga el módulo de aire.

- Reglas para detección de desastres:

```
; Detección de desastres
(defrule manejar-incendio
  ?zona-datos <- (zona (nombre ?zona) (tipo ?tipo-zona) (ocupacion ?ocupacion) (estado-desastre "activo"))
  =>
  (if (eq ?ocupacion 0) then
    ; Si la zona está vacía
    (if (eq ?tipo-zona "sensible") then
      (printout t "Incendio detectado en la zona " ?zona ". Zona sensible, sin ocupantes. Usando gas para apagarlo." crlf)
    else
      (printout t "Incendio detectado en la zona " ?zona ". Zona normal, sin ocupantes. Usando agua para apagarlo." crlf))
    (modify ?zona-datos (estado-desastre "normal"))) ; Cambiar el estado a normal tras apagar el incendio
  else
    ; Si la zona está ocupada, proceder a la evacuación
    (printout t "Incendio detectado en la zona " ?zona ". Evacuando ocupantes." crlf)

    ; Realiza la evacuación
    (modify ?zona-datos (ocupacion 0))
    (printout t "Usuarios evacuados de "?zona". " crlf)
    (if (eq ?tipo-zona "sensible") then
      (printout t "Zona sensible, ocupantes evacuados. Usando gas para apagarlo." crlf)
    else
      (printout t "Zona normal, ocupantes evacuados. Usando agua para apagarlo." crlf))
    (modify ?zona-datos (estado-desastre "normal"))) ; Actualiza ocupación a 0
    (printout t "Incendio controlado." crlf)
  )
)
```

Con la regla presentada se consigue manejar los casos de incendios en la simulación, cuando el sensor detecta que estado-desastre está activo lanza el primer if que detecta la presencia de personas en la sala afectada por el incendio, en caso de que esta sea nula, se actuará con agua o gas dependiendo de si el tipo de la sala es sensible o normal.

En el caso de que haya usuarios en la sala afectada, priorizará la evacuación de estos, una vez evacuados procederá a la extinción del incendio con la misma casuística que en el primer caso.

- Reglas para control de iluminación:

```
; Control de Iluminacion
(defrule control-iluminacion
  ?zona <- (zona (nombre ?nombre) (ocupacion ?ocupacion) (iluminacion ?estado-iluminacion))
  (test (or (and (> ?ocupacion 0) (eq ?estado-iluminacion "apagada"))
            (and (= ?ocupacion 0) (eq ?estado-iluminacion "encendida"))))
  =>
  (if (> ?ocupacion 0) then
    (modify ?zona (iluminacion "encendida"))
    (printout t "Alguien ha entrado en "?nombre". Encendiendo luces." crlf)
  else
    (modify ?zona (iluminacion "apagada"))
    (printout t "No queda nadie en "?nombre". Apagando luces." crlf)))
```

En esta regla lo que conseguimos es que cuando en una zona la ocupación sea cero la luz vaya a estar apagada y cuando haya gente la luz se encienda, y así tener un ahorro de energía.

La regla se activa cuando la ocupación sea mayor de 0 y el estado de iluminación como “apagada”, o viceversa. Cuando la regla esté activada, dependiendo del caso en el que nos encontremos se cambiará el estado de la luz a apagada/encendida.

- Reglas para control de Alimentación (Racks):

```
; Control de Alimentacion
(defrule control-voltaje
  ?rack <- (rack (id ?id) (voltaje ?voltaje))
  (test (or(< ?voltaje 210) (> ?voltaje 230)))
  =>
  (printout t "Voltaje inadecuado en el rack con id: " ?id ". Reiniciando rack." crlf)
  (modify ?rack (voltaje 220))
)
```

La regla es útil para controlar el voltaje de los racks y que este no se salga del rango estable definido entre 210 y 230 voltios. Cuando esto pasa, la regla se activa y se reinicia el rack modificando así su voltaje a la media (220v).

Hechos Iniciales

```
; Facts iniciales de zonas
(deffacts zonas-iniciales

  (zona (nombre "Pasillo") (nivel 1) (temperatura 20) (ocupacion 0) (tipo "normal") (estado-desastre "normal") (iluminacion "apagada"))
  (zona (nombre "Cafeteria") (nivel 1) (temperatura 21) (ocupacion 1) (tipo "normal") (estado-desastre "normal") (iluminacion "encendida"))
  (zona (nombre "Despacho") (nivel 2) (temperatura 21) (ocupacion 1) (tipo "normal") (estado-desastre "normal") (iluminacion "encendida"))
  (zona (nombre "Telco") (nivel 2) (temperatura 23) (ocupacion 1) (tipo "sensible") (estado-desastre "normal") (iluminacion "encendida"))
  (zona (nombre "Servidores") (nivel 3) (temperatura 23) (ocupacion 0) (tipo "sensible") (estado-desastre "normal") (iluminacion "apagada"))
  (zona (nombre "Zona de control") (nivel 3) (temperatura 22) (ocupacion 1) (tipo "sensible") (estado-desastre "normal") (iluminacion "encendida"))
  (zona (nombre "Zona de Alimentacion (SAI)") (nivel 3) (temperatura 22) (ocupacion 0) (tipo "sensible") (estado-desastre "normal") (iluminacion "apagada")))
```

En estos hechos se definen las zonas de la simulación con sus valores iniciales.

```
; Facts iniciales de usuarios
(deffacts control-de-accesos

  (usuario (nombre "Directora María del Carmen") (nivel-acceso 3))
  (usuario (nombre "Controlador Juan Carlos (JC)") (nivel-acceso 3))
  (usuario (nombre "Especialista en redes Martín") (nivel-acceso 2))
  (usuario (nombre "Limpiadora Estela") (nivel-acceso 1))
  (usuario (nombre "Camarero Iván Luis") (nivel-acceso 1)))
```

Estos hechos definen los usuarios que estarán en el sistema y su nivel de acceso a zonas.


```

; Facts iniciales del módulo de climatización
(deffacts inicial-modulo-climatizacion
  (modulo-aire (sala "Pasillo") (temperatura-objetivo 22) (estado "apagado"))
  (modulo-aire (sala "Cafeteria") (temperatura-objetivo 22) (estado "apagado"))
  (modulo-aire (sala "Despacho") (temperatura-objetivo 22) (estado "apagado"))
  (modulo-aire (sala "Teleco") (temperatura-objetivo 22) (estado "apagado"))
  (modulo-aire (sala "Servidores") (temperatura-objetivo 22) (estado "apagado"))
  (modulo-aire (sala "Zona de control") (temperatura-objetivo 22) (estado "apagado"))
  (modulo-aire (sala "Zona de Alimentacion (SAI)") (temperatura-objetivo 22) (estado "apagado")))

```

Aquí se definen los módulos de aire de cada sala, la temperatura objetivo de cada uno es modificable por si en algún momento una sala necesita una climatización diferente a otras.

```

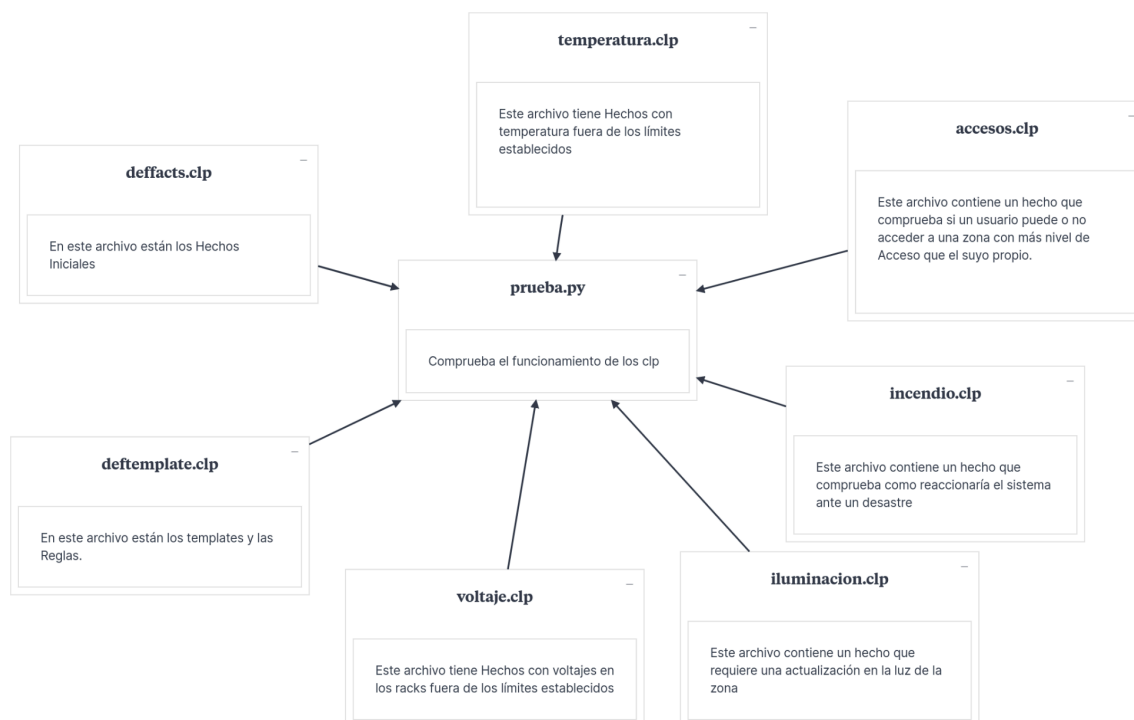
; Facts iniciales de racks
(deffacts racks-iniciales
  (rack (id 1) (voltaje 220))
  (rack (id 2) (voltaje 222)))

```

Por último, se definen los racks con su id y voltaje propio.

Funcionamiento

Nuestro programa gira en torno al programa prueba.py



En prueba.py se cargan todos los .clp

Primero cargamos nuestra base de hechos iniciales (deffacts.clp) y nuestra base de templates y reglas (deftemplate.clp).

Después de eso realizamos un print de los usuarios y su nivel de acceso para ver que se han cargado de forma correcta y hacemos distintas simulaciones para comprobar el funcionamiento de las reglas.

Las simulaciones tienen todas la misma estructura, cargar los hechos de un .clp nuevo (accesos, incendio...etc) después de eso hacemos un reset para pasar esos hechos a la memoria activa y finalmente un run para comprobar que se ejecuten las reglas de forma adecuada frente a hechos que las satisfagan.

Simulaciones:

Accesos (accesos.clp):

Aquí se comprueba que pasa cuando un usuario intenta acceder a una zona:

```
----- SIMULACIÓN DE ACCESOS -----  
Acceso denegado para Especialista en redes Martín a la zona Zona de Alimentacion (SAI). Nivel de acceso insuficiente.  
Acceso permitido para Limpiadora Estela a la zona Pasillo. Nivel de acceso suficiente.  
No queda nadie en Cafeteria. Apagando luces.  
Alguien ha entrado en Pasillo. Encendiendo luces.
```

Temperatura (temperatura.clp):

Aquí se comprueba que pasa cuando la temperatura de una zona sale de los límites establecidos

```
----- SIMULACIÓN DE TEMPERATURA -----  
  
Temperatura en Zona de control no ideal, aire encendido  
Ajustando temperatura en Zona de control de 35 a 34  
Ajustando temperatura en Zona de control de 34 a 33  
Ajustando temperatura en Zona de control de 33 a 32  
Ajustando temperatura en Zona de control de 32 a 31  
Ajustando temperatura en Zona de control de 31 a 30  
Ajustando temperatura en Zona de control de 30 a 29  
Ajustando temperatura en Zona de control de 29 a 28  
Ajustando temperatura en Zona de control de 28 a 27  
Ajustando temperatura en Zona de control de 27 a 26  
Ajustando temperatura en Zona de control de 26 a 25  
Ajustando temperatura en Zona de control de 25 a 24  
Ajustando temperatura en Zona de control de 24 a 23  
Ajustando temperatura en Zona de control de 23 a 22  
Temperatura en Zona de control ideal, aire apagado  
  
Temperatura en Pasillo no ideal, aire encendido  
Ajustando temperatura en Pasillo de 18 a 19  
Ajustando temperatura en Pasillo de 19 a 20  
Ajustando temperatura en Pasillo de 20 a 21  
Ajustando temperatura en Pasillo de 21 a 22  
Temperatura en Pasillo ideal, aire apagado
```

Incendio (incendio.clp):

Aquí se comprueba que pasa en caso de que un desastre esté en “activo”

```
----- SIMULACIÓN DE INCENDIO -----  
Incendio detectado en la zona Teleco. Evacuando ocupantes.  
Usuarios evacuados de Teleco.  
Zona sensible, ocupantes evacuados. Usando gas para apagarlo.  
Incendio controlado.  
No queda nadie en Teleco. Apagando luces.
```

Iluminación (iluminacion.clp):

Aquí comprobamos el funcionamiento de las luces en función de los cambios de ocupación de las zonas

```
----- SIMULACIÓN DE ILUMINACIÓN -----  
Alguien ha entrado en Teleco. Encendiendo luces.
```

Voltaje (voltaje.clp):

Aquí comprobamos que en el momento en que un Rack obtiene valores fuera de los rangos se arregle.

```
----- SIMULACIÓN DE VOLTAJE -----  
Voltaje inadecuado en el rack con id: 3. Reiniciando rack.
```

Manual de Uso

Requisitos del Programa

- 1.Tener instalado **Python**
- 2.Instalar la librería **Clipsy** (En linux es posible que tenga que crear un entorno virtual para poder proceder con la instalación)

Uso del Programa

- 1.Ubícate en la carpeta del programa.
- 2.Ejecuta el archivo **prueba.py**.

Conclusiones

A lo largo de este proyecto, hemos logrado desarrollar un sistema experto funcional que simula el control de un centro de datos, abarcando diversas áreas críticas como el acceso, la climatización, la detección de incendios, el control de iluminación y el monitoreo del voltaje en racks.

Durante el desarrollo, aprendimos a manejar la complejidad de las reglas en CLIPS y a trabajar con Clipspy, lo cual supuso un reto para nosotros. Los múltiples errores nos llevaron a simplificar las reglas para evitar complicaciones y mejorar la eficiencia del sistema.

A pesar de que hemos conseguido satisfacer los objetivos que nos hemos propuesto (Alcance del sistema) podríamos haber conseguido un sistema más sólido y con más cosas a tener en cuenta (Humedad, Calefacción global...etc).

En resumen, el desarrollo nos ha permitido aprender acerca de lo complicado de las reglas y la importancia de definir bien lo que necesita nuestro sistema antes de comenzar a programarlo.

Bibliografía

https://moovi.uvigo.gal/pluginfile.php/1841267/mod_assign/intro/SIEX_practica1.html?time=1728228615022

<https://moovi.uvigo.gal/pluginfile.php/1567599/course/section/152810/Guia%20de%20la%20Practica1.pdf>

https://moovi.uvigo.gal/pluginfile.php/1567599/course/section/152810/SIEX_resumen_clips.pdf

<https://www.clipsrules.net/>

<https://elvex.ugr.es/decsai/intelligent/workbook/ai/CLIPS.pdf>

<https://clipspy.readthedocs.io/en/latest/>

Problemas Encontrados en el desarrollo

Durante el desarrollo nos hemos encontrado con distintos problemas siendo el más destacable de ellos la complejidad a la hora de crear las reglas, esto es debido a la complejidad de las mismas, puesto que al modificar valores se llaman a otras reglas y nos encontramos en muchas ocasiones con bucles infinitos.

Debido a estos problemas decidimos simplificar las reglas para poder abordarlas de una forma más sencilla y dejamos de lado ciertos sistemas como la humedad en la zona o la calefacción global.

Al principio tuvimos bastantes dificultades con la sintaxis de Clips pero este problema se fue solucionando conforme avanzaba el proyecto.