

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# CSUOptimizer: Optimización de apuntados en el proyecto EMIR

**Pedro Orlando Hernández Martín**

PROYECTO DE FIN DE CARRERA,  
INGENIERÍA INFORMÁTICA  
La Laguna, 16 Julio 2013

## IAC

[Introducción](#)[Unidad de rejillas  
configurable](#)[CSUOptimizer:  
Optimizando  
apuntados](#)[Resultados](#)[Conclusiones](#)

## Sedes

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



## GTC

Introducción

Unidad de rejillas  
configurable

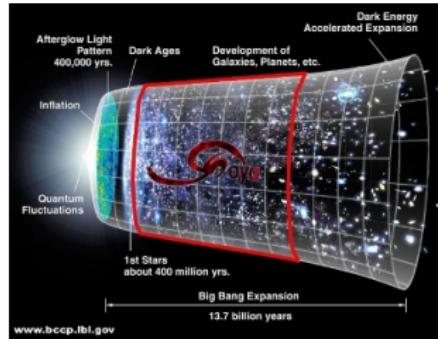
CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Proyecto GOYA



## GOYA: Galaxy Origins and Young Assembly

- Objetivo: entender cómo se forman y evolucionan las galaxias observando sus propiedades en una época de máximo crecimiento
- Investiga las propiedades de las galaxias más lejanas
- A distancias de 8000–13000 millones de años luz
- Llevará a cabo la primera exploración sistemática de la población de galaxias poco después del nacimiento del universo

# El Proyecto EMIR

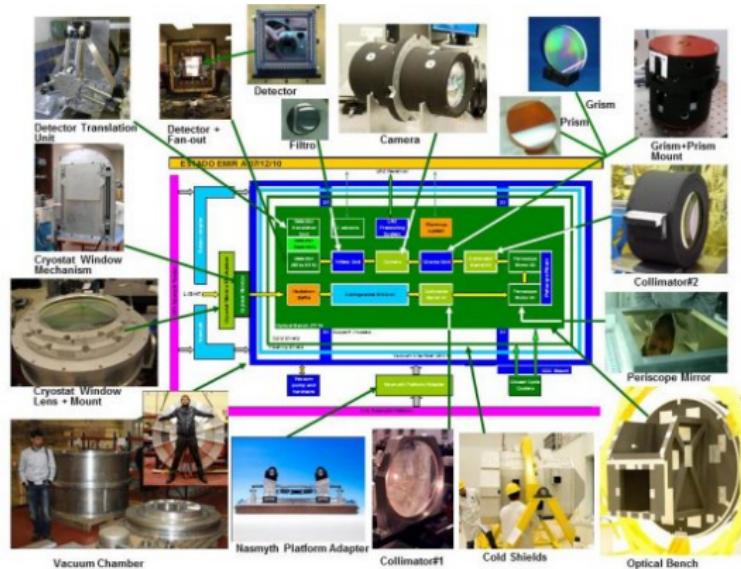
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



## EMIR: Espectrógrafo Multiobjeto Infrarrojo

EMIR es un cámara de gran campo y espectrógrafo de resolución intermedia en el infrarrojo cercano para el telescopio GTC

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## 1 Introducción

## 2 Unidad de rejillas configurable

## 3 CSUOptimizer: Optimizando apuntados

## 4 Resultados

## 5 Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## 1 Introducción

## 2 Unidad de rejillas configurable

## 3 CSUOptimizer: Optimizando apuntados

## 4 Resultados

## 5 Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## 1 Introducción

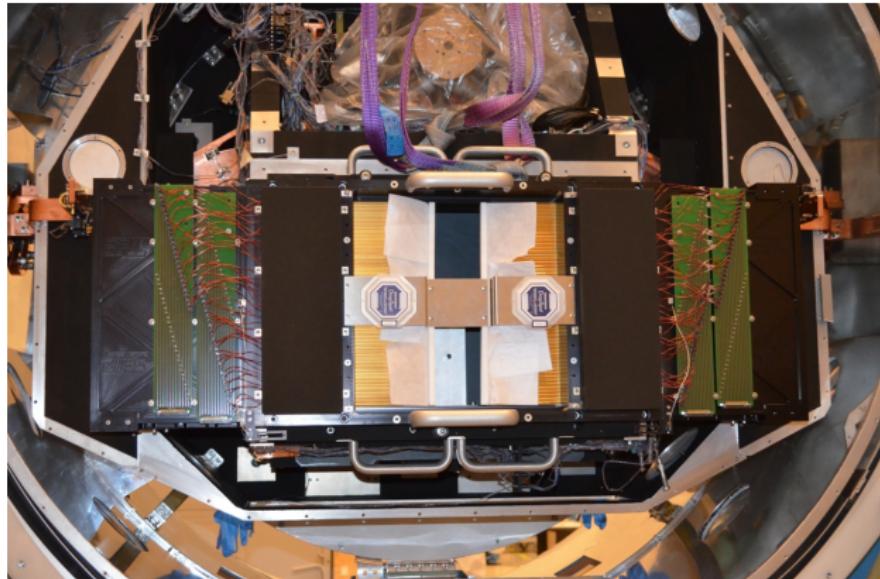
## 2 Unidad de rejillas configurable

## 3 CSUOptimizer: Optimizando apuntados

## 4 Resultados

## 5 Conclusiones

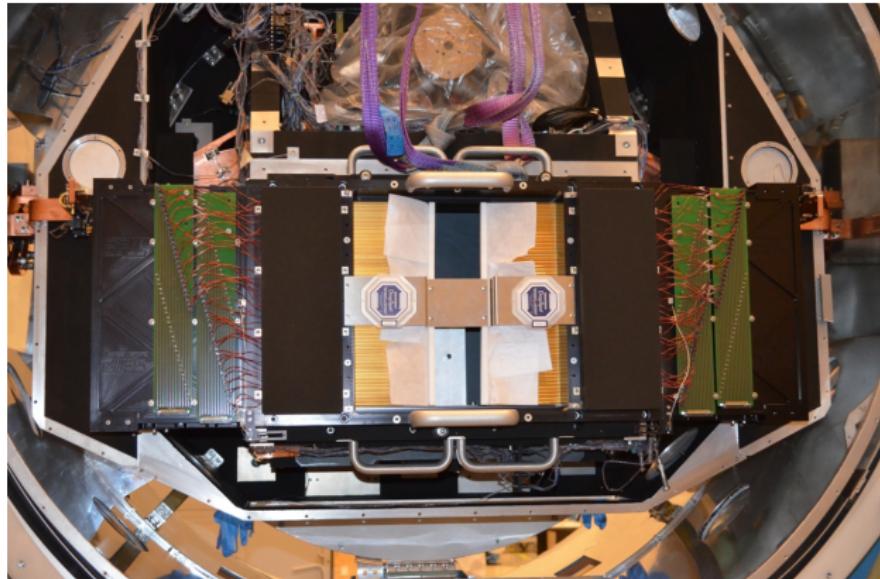
# CSU: Configurable Split Unit



## CSU

- Dimensiones →  $240 \times 400$  arcosegundos

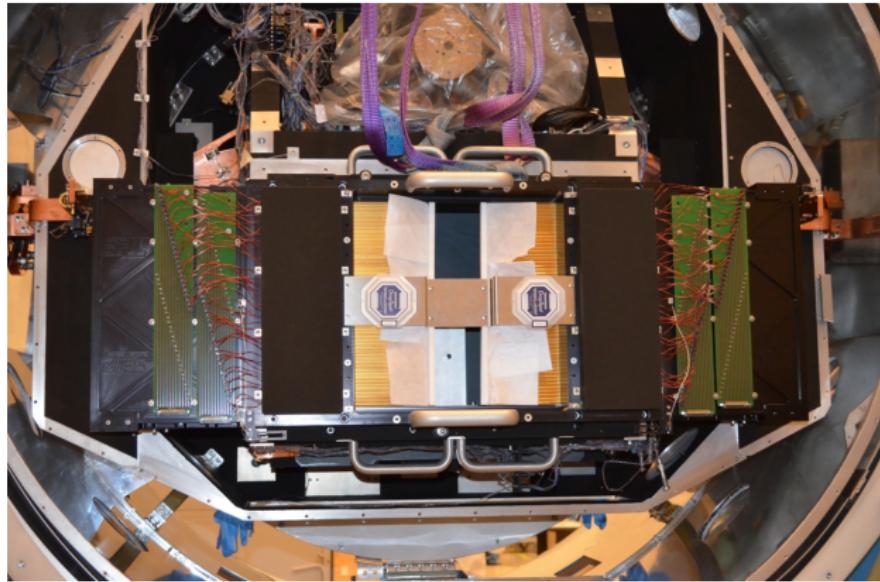
# CSU: Configurable Split Unit



## CSU

- Dimensiones →  $240 \times 400$  arcosegundos
- 55 pares de barras

# CSU: Configurable Split Unit

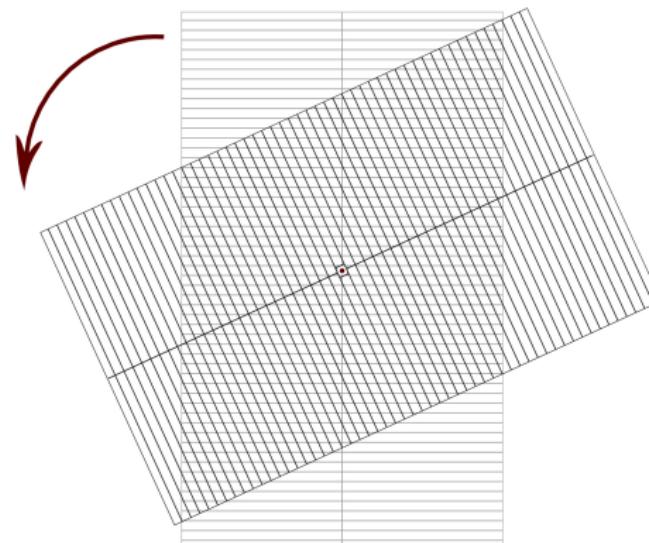
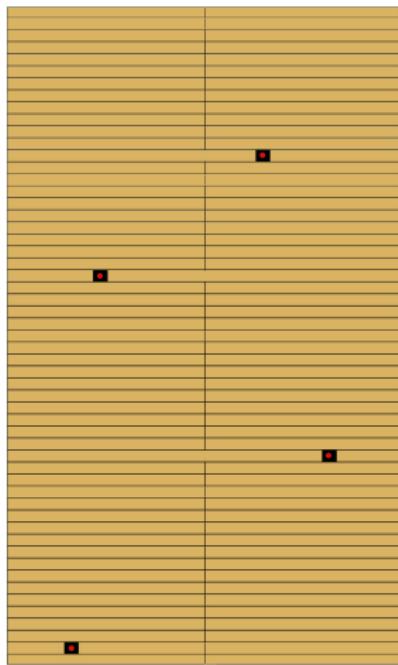


## CSU

- Dimensiones →  $240 \times 400$  arcosegundos
- 55 pares de barras
- Rotación

# Apuntado - Representación gráfica

Introducción  
Unidad de rejillas  
configurable  
CSUOptimizer:  
Optimizando  
apuntados  
Resultados  
Conclusiones



# Problema



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Descripción

- Número de objetos a observar >> número de barras

# Problema



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Descripción

- Número de objetos a observar >> número de barras
- Minimizar el tiempo de observación es crítico

# Problema



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Descripción

- Número de objetos a observar >> número de barras
- Minimizar el tiempo de observación es crítico
- Configuración compleja

# Objetivos

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

Una aplicación que:

- Resuelva el problema

# Objetivos

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Una aplicación que:

- Resuelva el problema
- Minimice el tiempo de respuesta

# Objetivos

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Una aplicación que:

- Resuelva el problema
- Minimice el tiempo de respuesta
- Contemple prioridades

# Objetivos

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Una aplicación que:

- Resuelva el problema
- Minimice el tiempo de respuesta
- Contemple prioridades
- Contemple Beam Switching

# Problema

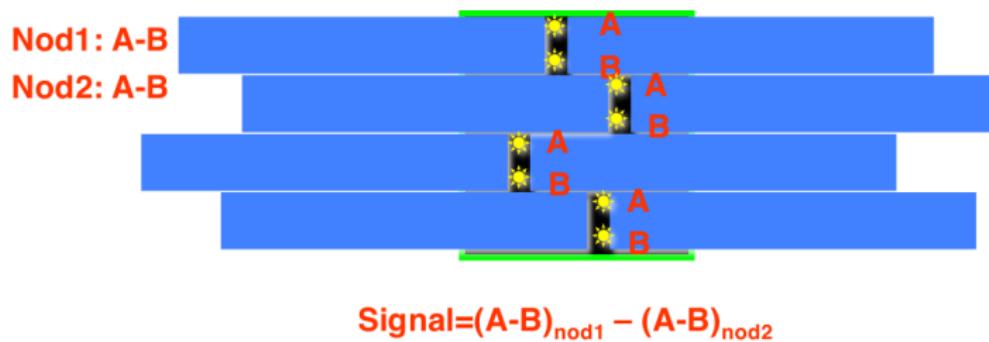
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



## Beam switching

- Método para obtener mediciones más precisas
- Se mide la luz procedente del objeto y la de la bóveda celeste
- Se mueve ligeramente la posición del medidor y se repite la medida

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Frentes de actuación

- Clustering
- Algoritmo Constructivo

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## DBScan

```
DBSCAN( $D$ ,  $\text{eps}$ ,  $\text{MinPts}$ ) {
     $C = 0$ 
    For(cada punto  $P$  no visitado en el cjto de datos  $D$ ) {
        marcar  $P$  como visitado
         $NeighborPts = \text{regionQuery}(P, \text{eps})$ 
        If(sizeof( $NeighborPts$ ) <  $\text{MinPts}$ )
            marcar  $P$  como RUIDO
        else {
             $C = \text{siguiente cluster}$ 
            expandCluster( $P$ ,  $NeighborPts$ ,  $C$ ,  $\text{eps}$ ,  $\text{MinPts}$ )
        }
    }
}
```

# Clustering

## DBSCAN

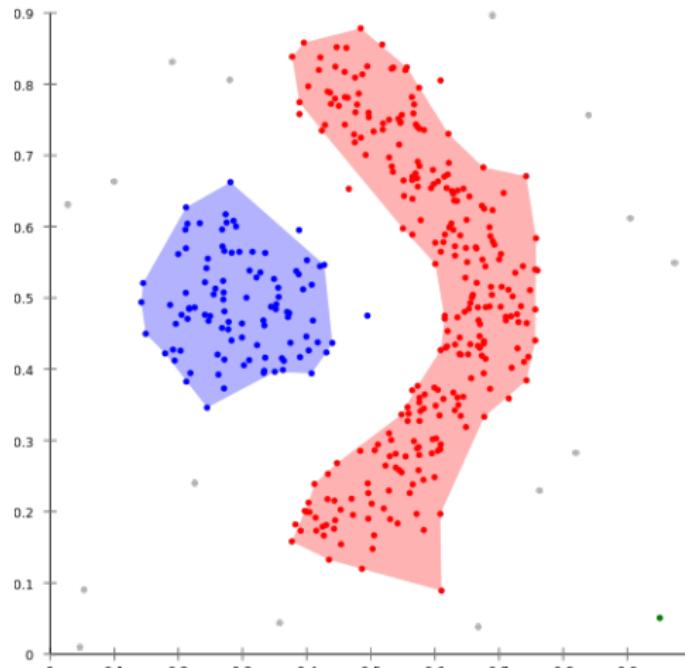
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Clustering

## DBSCAN

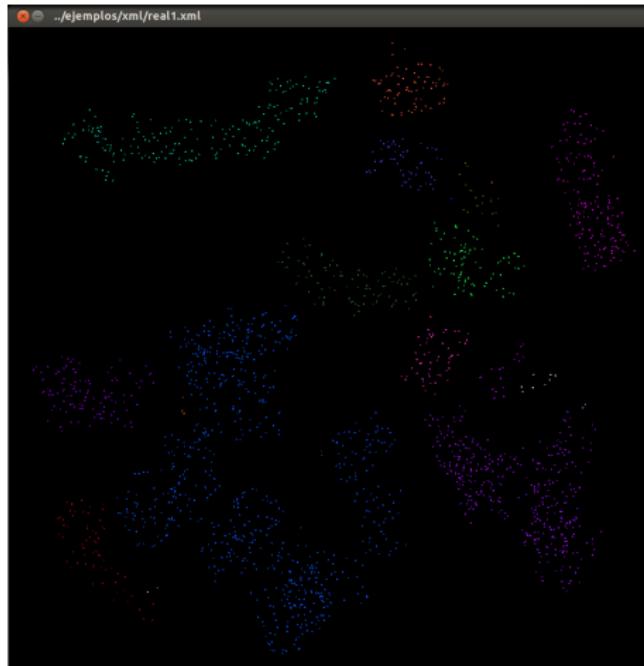
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Algoritmo Constructivo

```
For (Cada tipo de ordenacion) {  
    ordenar los objetos  
    While (queden objetos por cubrir) {  
         $p \leftarrow$  siguiente punto de la lista no eliminado  
         $C =$  mejor CSU  $\in$  crearApuntados( $p$ , puntos)  
        puntos -=  $\forall$  puntos  $\in C$   
        Solucion[tipoorden] = Solucion[tipo orden]  $\cup C$   
    }  
}  
Sol = min(Solucion[tipo orden])
```

# Algoritmo Constructivo

## Fase 1: Obtención de objetos

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

```
int estaDentro(const Element &p) const {
    static double new_x, new_y, zone;
    if (estaDentro2(p)) {
        testeo(p, new_x, new_y);
        zone = sqrt((Ax - new_x) * (Ax - new_x) + (Ay - new_y) * (Ay - new_y));
        barra = (int)(zone / DIST_BARRAS);
        if (barra == NUM_BARRAS)
            return 0;
        rango = zone - (barra * DIST_BARRAS);
        distancia = sqrt((p.getx() - new_x) * (p.getx() - new_x) + (p.gety() - new_y) * (p.gety() - new_y));
        if (distancia > 2*ANCHO)
            return 0;
        if (barras_ocupadas[barra])
            return p_potencial(rango, barra);
    }
    return 0;
}
```

# Algoritmo Constructivo

## Fase 1: Obtención de objetos

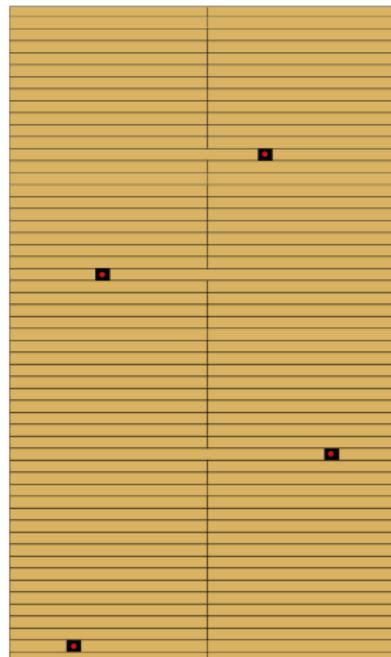
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Fase 2: Mejora

```
For(Todas las rotaciones posibles) {
    Crear CSU con centro en en el objeto
    llenar_con_puntos(CSU, lista_objetos)
    While(numero de puntos en CSU cambie) {
        movimiento de mejora arriba
        llenar_con_puntos(CSU, puntos)
        movimiento de mejora abajo
        llenar_con_puntos(CSU, puntos)
        movimiento de mejora izquierda
        llenar_con_puntos(CSU, puntos)
        movimiento de mejora derecha
        llenar_con_puntos(CSU, puntos)
    }
    posibles = posibles ∪ CSU
}
```

# Algoritmo Constructivo

## Fase 2: Mejora

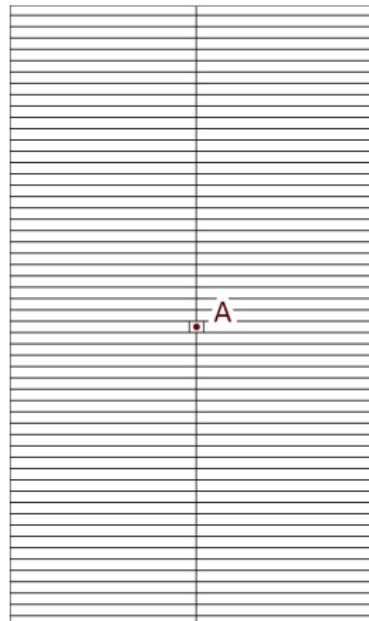
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Algoritmo Constructivo

## Fase 2: Mejora

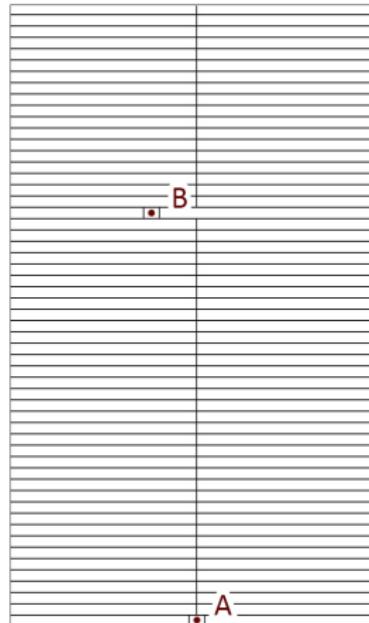
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Algoritmo Constructivo

## Fase 2: Mejora

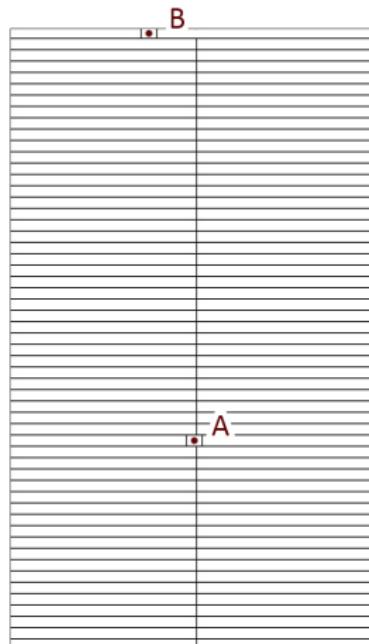
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Algoritmo Constructivo

## Fase 2: Mejora

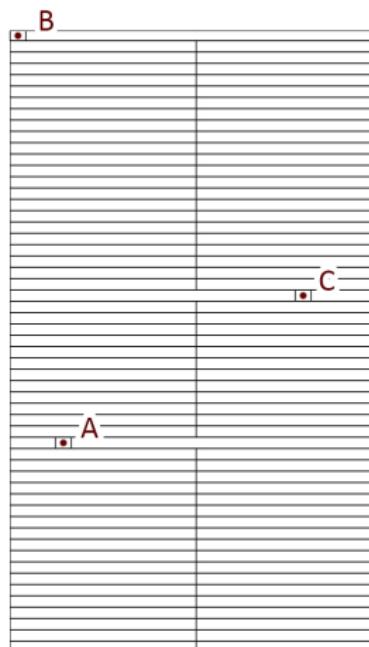
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Algoritmo Constructivo

## Fase 2: Mejora

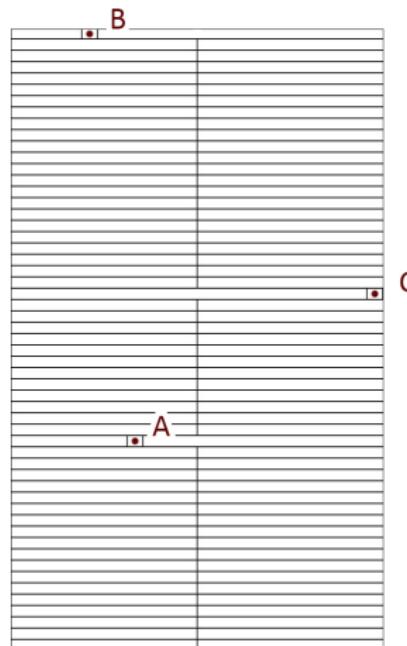
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Obtención de objetos

## Problema - Colisión

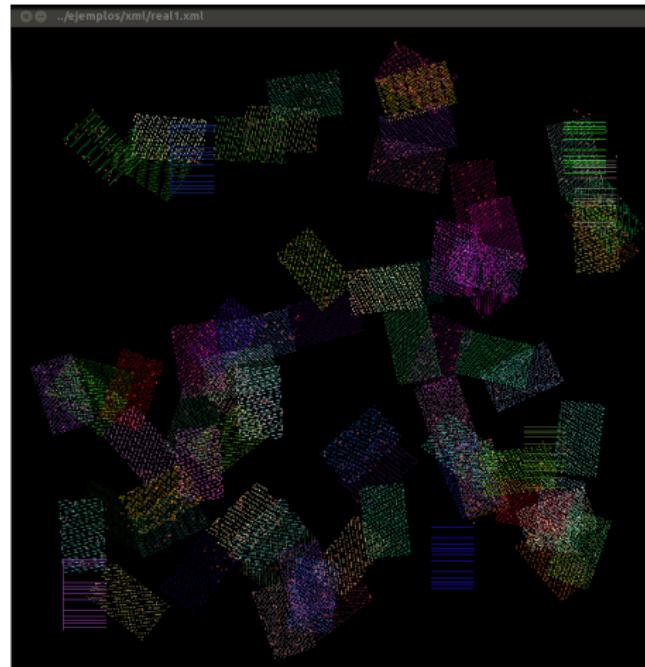
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Obtención de objetos

## Problema - Colisión

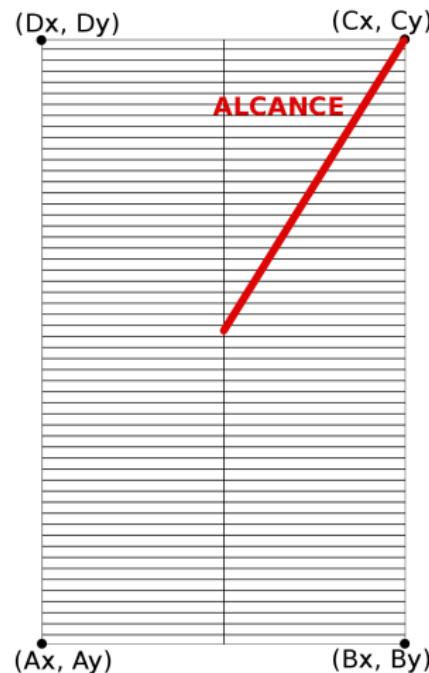
Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



# Reducción de colisiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

```
listaini = resultado
While(Mientras se mejore el resultado) {
    listafin = los que no colisionan con ninguno.
    While(queden objetos por cubrir) {
        CSU C ← primer apuntado ∈ listaini
        subcto_puntos = puntos de C
        Quitar C de listaini
        While(queden CSUs Q por mirar en listaini) {
            If(Q colisiona con C) {
                subcto_puntos+= puntos de Q
                Quitar Q de listaini
            }
        }
        If(colision innecesaria) {
            Obtener_nuevos_apuntados(subcto_puntos)
            If(consigue mejorar)
                listaini+= nuevos apuntado
            else {
                Introducir en listaini los que se quitaron
                Poner en listafin el apuntado C
            }
        }
    }
    listaini = listafin
```

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## 1 Introducción

## 2 Unidad de rejillas configurable

## 3 CSUOptimizer: Optimizando apuntados

## 4 Resultados

## 5 Conclusiones

[Introducción](#)[Unidad de rejillas  
configurable](#)[CSUOptimizer:  
Optimizando  
apuntados](#)[Resultados](#)[Conclusiones](#)

## Entrada/Salida

```
./csuoptimizer [opciones] entrada.xml
```

```
<Observables [width='Ancho_cielo', height='Alto_cielo']>
    <Element id='Identificador'>
        <X>PosX</X>
        <Y>PosY</Y>
        <Prioridad>[1..99]</Prioridad>
    </Element>
    <!-- ... -->
</Observables>
```

[Introducción](#)[Unidad de rejillas  
configurable](#)[CSUOptimizer:  
Optimizando  
apuntados](#)[Resultados](#)[Conclusiones](#)

## Entrada/Salida

```
<Apuntado>
    <X>ValorX</X>
    <Y>ValorY</Y>
    <PA>AnguloPosicion</PA>
    <Configuracion>
        <Barra>Posicion1</Barra>
        <Barra>Posicion2</Barra>
            <! -- ... -->
        <Barra>Posicion54</Barra>
        <Barra>Posicion55</Barra>
    </Configuracion>
</Apuntado>
```

[Introducción](#)[Unidad de rejillas  
configurable](#)[CSUOptimizer:  
Optimizando  
apuntados](#)[Resultados](#)[Conclusiones](#)

## Opciones

- Uso del DBScan (No)

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Opciones

- Uso del DBScan (No)
- Elegir número de rotaciones (20)

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Opciones

- Uso del DBScan (No)
- Elegir número de rotaciones (20)
- Elegir tipo de salida (XML)

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Opciones

- Uso del DBScan (No)
- Elegir número de rotaciones (20)
- Elegir tipo de salida (XML)
- Tipo de barra (Normal)

# Tipos de barras

Introducción

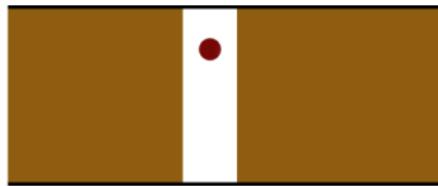
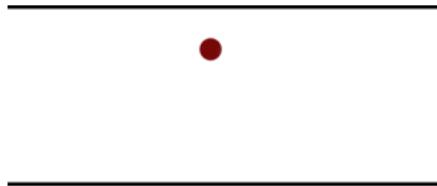
Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Barras normales



# Tipos de barras

Introducción

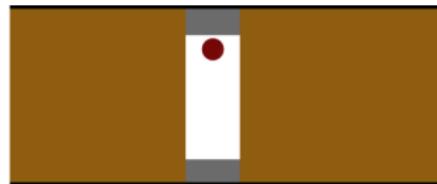
Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Barras sin bordes



# Tipos de barras

Introducción

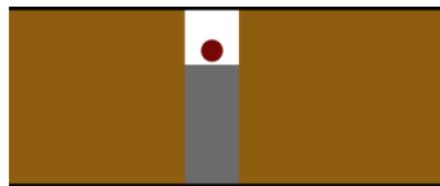
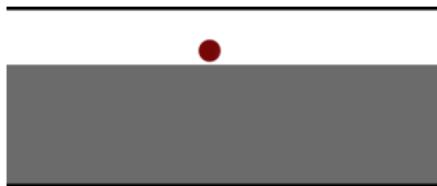
Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Barras con Beam Switching



# Tipos de barras

Introducción

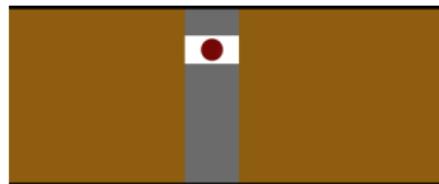
Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Barras con ambas opciones



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## 1 Introducción

## 2 Unidad de rejillas configurable

## 3 CSUOptimizer: Optimizando apuntados

## 4 Resultados

## 5 Conclusiones

# Resultados

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones



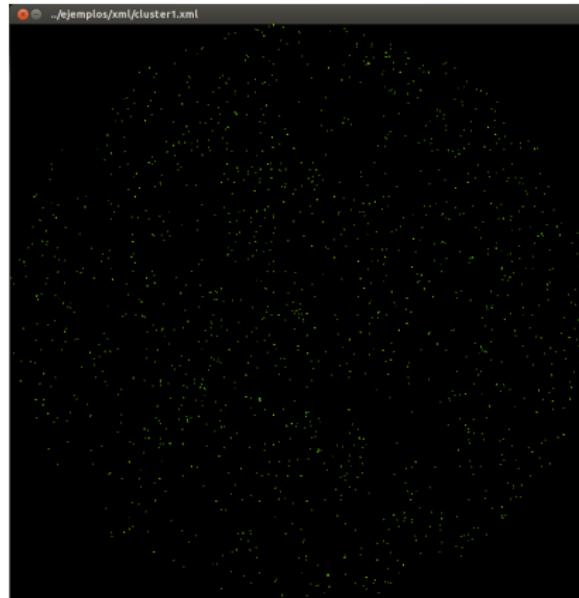
## Allegro

- Librería libre y de código abierto para la programación de videojuegos desarrollada en lenguaje C
- Funciones para gráficos, manipulación de imágenes, texto, sonidos, etc.

# Resultados

## Caso Real: cluster1.xml

- 1398 objetos
- Distribución concéntrica



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

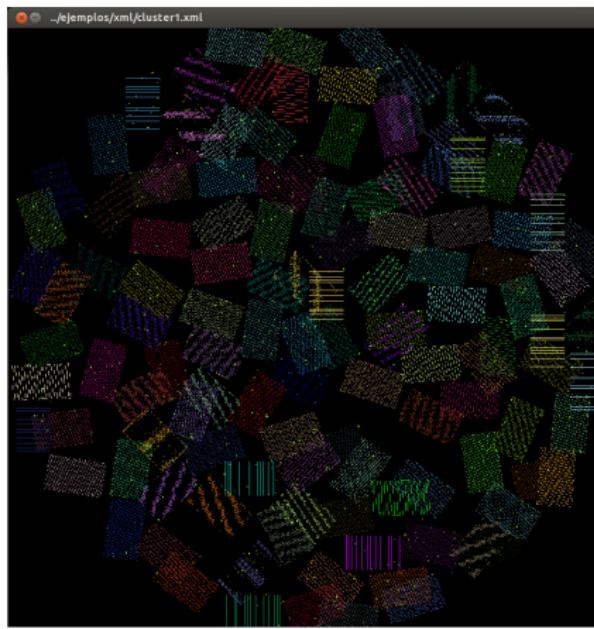
Resultados

Conclusiones

# Resultados

## Caso Real: cluster1.xml

- 1398 objetos
- Distribución concéntrica



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# Resultados

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

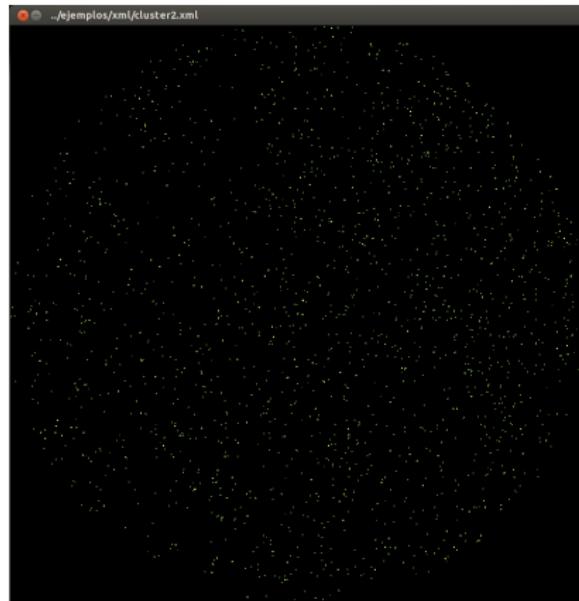
## Caso Real: cluster1.xml

RESULTADOS	Apuntados	Tiempo
Sin opciones	123	45.04''
Sin bordes	138	45.05''
Beam Switching	180	1' 29.35''
Sin bordes + Beam S.	218	1' 59.17''

# Resultados

## Caso Real: cluster2.xml

- 1842 objetos
- Distribución concéntrica



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

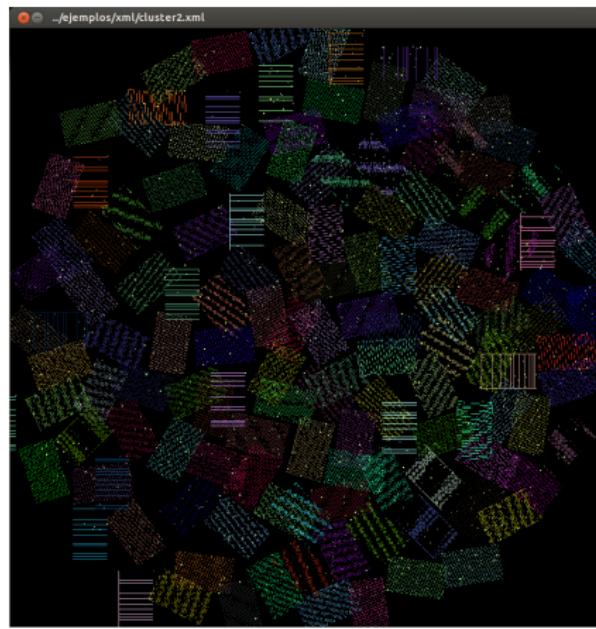
Resultados

Conclusiones

# Resultados

## Caso Real: cluster2.xml

- 1842 objetos
- Distribución concéntrica



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# Resultados

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

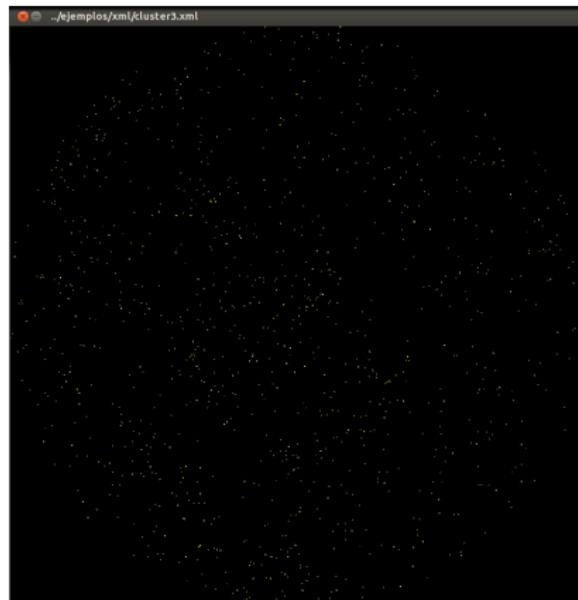
## Caso Real: cluster2.xml

RESULTADOS	Apuntados	Tiempo
Sin opciones	138	1' 57.42''
Sin bordes	152	2' 1.68''
Beam Switching	213	2' 19.05''
Sin bordes + Beam S.	263	4' 19.05''

# Resultados

## Caso Real: cluster3.xml

- 1074 objetos
- Distribución concéntrica



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

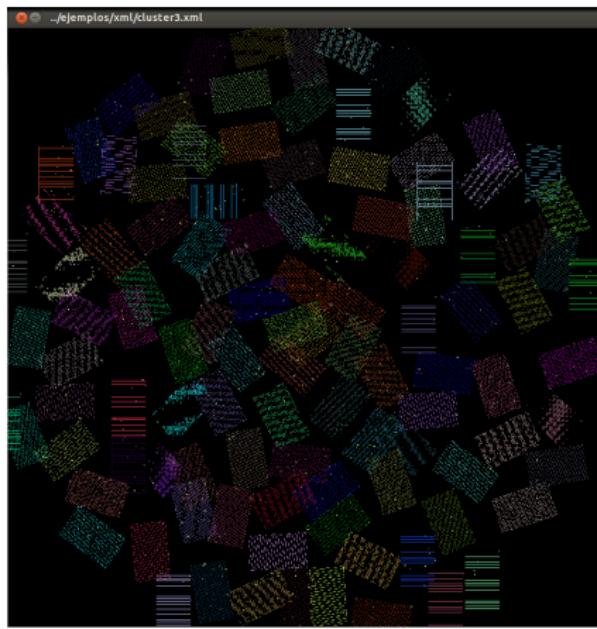
Resultados

Conclusiones

# Resultados

## Caso Real: cluster3.xml

- 1074 objetos
- Distribución concéntrica



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# Resultados

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

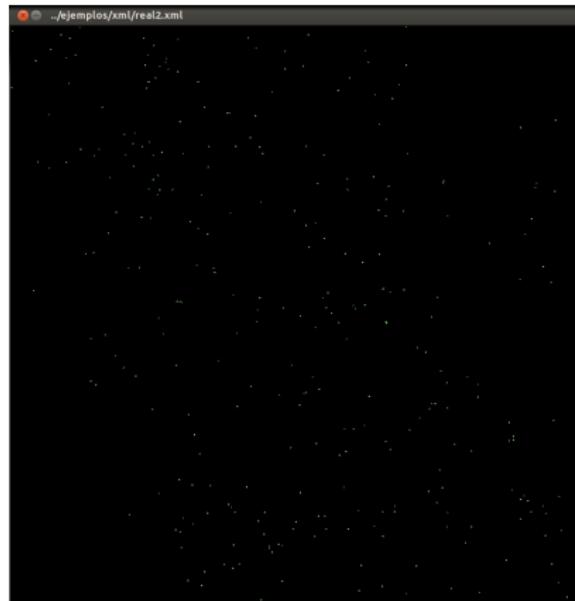
## Caso Real: cluster3.xml

RESULTADOS	Apuntados	Tiempo
Sin opciones	109	31.09''
Sin bordes	117	38.57''
Beam Switching	153	1' 4.49''
Sin bordes + Beam S.	180	1' 16.68''

# Resultados

## Caso Real: real2.xml

- 297 objetos
- Distribución aleatoria dispersa



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

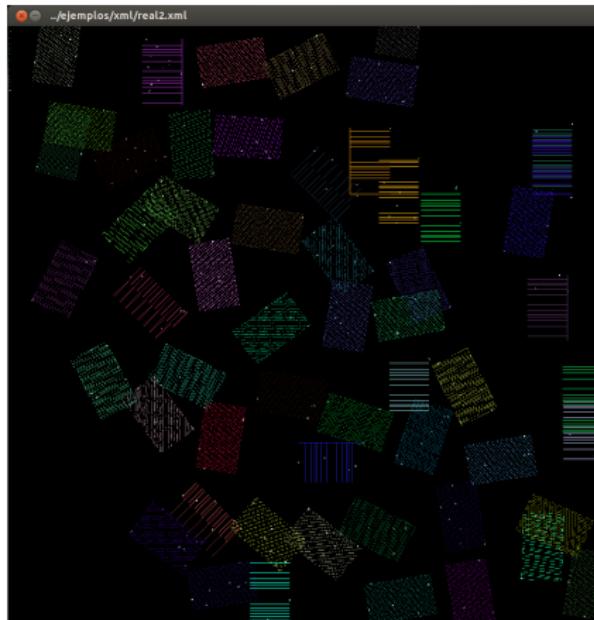
Resultados

Conclusiones

# Resultados

## Caso Real: real2.xml

- 297 objetos
- Distribución aleatoria dispersa



Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# Resultados

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Caso Real: real2.xml

RESULTADOS	Apuntados	Tiempo
Sin opciones	57	3.85 ''
Sin bordes	56	3.91 ''
Beam Switching	70	6.24 ''
Sin bordes + Beam S.	79	7.08 ''

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# Volvamos a la prueba

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## 1 Introducción

## 2 Unidad de rejillas configurable

## 3 CSUOptimizer: Optimizando apuntados

## 4 Resultados

## 5 Conclusiones

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Logros

- Investigación sobre posibles métodos para resolver el problema

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Logros

- Investigación sobre posibles métodos para resolver el problema
- Desarrollo de una aplicación plenamente operativa

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Logros

- Investigación sobre posibles métodos para resolver el problema
- Desarrollo de una aplicación plenamente operativa
- Software documentado como punto de partida de desarrollos futuros

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Logros

- Investigación sobre posibles métodos para resolver el problema
- Desarrollo de una aplicación plenamente operativa
- Software documentado como punto de partida de desarrollos futuros
- Documento técnico de la aplicación desarrollada

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Dificultades

- La mejora de la calidad de las soluciones obtenidas

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Dificultades

- La mejora de la calidad de las soluciones obtenidas
- La reducción de los tiempos de ejecución

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Dificultades

- La mejora de la calidad de las soluciones obtenidas
- La reducción de los tiempos de ejecución
- Estudio y selección de algoritmos para el problema planteado

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Dificultades

- La mejora de la calidad de las soluciones obtenidas
- La reducción de los tiempos de ejecución
- Estudio y selección de algoritmos para el problema planteado
- Adaptarnos al lenguaje usual de los astrónomos

# Conclusiones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Dificultades

- La mejora de la calidad de las soluciones obtenidas
- La reducción de los tiempos de ejecución
- Estudio y selección de algoritmos para el problema planteado
- Adaptarnos al lenguaje usual de los astrónomos
- La comprensión de las características intrínsecas al problema

# Trabajos futuros

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Posibles líneas de actuación

- Estudio de calidad de soluciones

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Posibles líneas de actuación

- Estudio de calidad de soluciones
- Margen de mejora en tiempos de respuesta

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

## Possibles línies de actuación

- Estudio de calidad de soluciones
- Margen de mejora en tiempos de respuesta
- Interfaz de usuario

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# **Gracias por su atención**

Introducción

Unidad de rejillas  
configurable

CSUOptimizer:  
Optimizando  
apuntados

Resultados

Conclusiones

# CSUOptimizer: Optimización de apuntados en el proyecto EMIR

**Pedro Orlando Hernández Martín**

PROYECTO DE FIN DE CARRERA,  
INGENIERÍA INFORMÁTICA  
La Laguna, 16 Julio 2013