



## Escuela KiCad

### Día 2:

- Editores de símbolos y módulos.
- Editor de Esquemático: Práctica intermedia.
- Editor de PCB: Práctica multicapa simple.
- Uso de GIT.

Este material es una adaptación del segundo día del curso dictado por INTI-CMNB y solicitado por ACSE, de autor **Diego Brengi** y **Noelia Scotti** con aportes de **Diego Alamon**

Adaptación de **Diego Brengi**, para la Carrera de Especialización en Sistemas Embebidos.

# Ejercicio 2

## Profundizando en el uso de KiCad

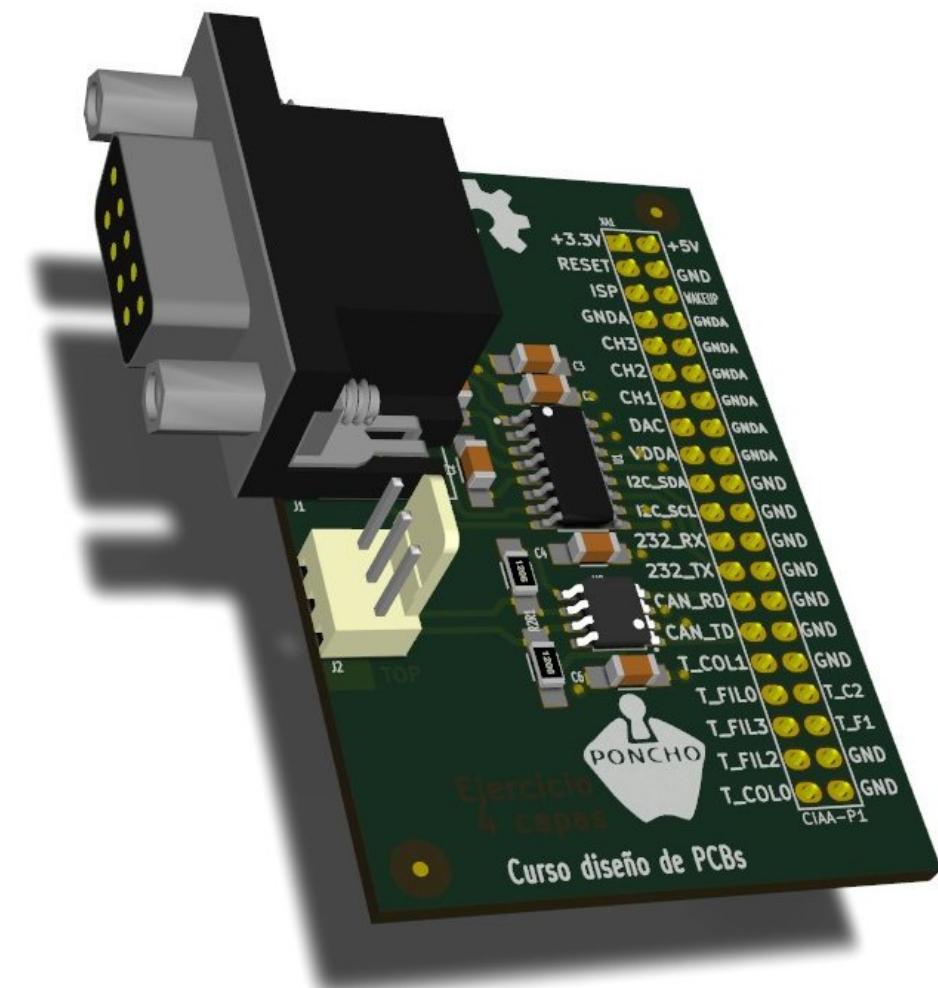


# Ejercicio 2 - Inicio

En este ejercicio iremos viendo un poco más en profundidad las opciones de KiCad.

El ejercicio consiste en un circuito de cuatro capas simple, pero se enfocará en tareas y métodos que son utilizados de igual forma en un diseño de mediana complejidad y en circuitos multicapa:

- Uso de jerarquía de hojas.
- Uso de etiquetas y buses.
- Funcionamiento del ERC.
- Creación de símbolos y footprints de biblioteca.
- Mayor profundidad en las opciones del Pcbnew.
- Zonas de cobre.
- Vista 3D.
- Generación y verificación de Gerbers.



- 1) Crear un proyecto “Ejercicio2”.
- 2) Guardar.

# Ejercicio 2 - Planificación de directorios

Nuestro diseño tendrá una estructura de directorios planificada previamente. No hay una regla estricta para esta organización, pero es muy conveniente tenerla y mantenerla en todos nuestros diseños.

Sugerencia:

## Ejercicio2

**libs**: Biblioteca de símbolos de nuestro diseño.

**ej2.pretty**: Footprints.

**ej2.3dshapes**: Modelos 3D.

**tools**: Herramientas complementarias.

**gerber**: Archivos gerber.

**doc**: Documentación y licencia.

**datasheets**: Hojas de datos relevantes.

**1) Crear la estructura de directorios.**



# Ejercicio 2

## Editor de esquemáticos

# Esquemático - El rótulo

Completar adecuadamente el rótulo para una documentación formal y prolja:

- El campo fecha no es automático, conviene actualizarlo luego de cambios significativos.
- El campo revisión se debe actualizar por lo menos luego de cambios mayores.
- El campo Company siempre debe reflejar la institución/empresa o sino el autor.
- Para el caso de proyectos abiertos, es conveniente mencionar la licencia, o indicar donde se encuentra.
- Para el caso de varios autores se puede indicar donde encontrar el listado de autores.



**Tip:** Utilizar la opción de exportar a las demás hojas para los campos comunes. El título no es común a todos ya que debe reflejar el contenido de cada hoja particular.

**Tip:** Recordar que el PCB también tiene rótulo.

# Ejercicio 2 - Configuración de bibliotecas

Para este ejercicio intentaremos solamente utilizar la biblioteca de símbolos provista.

**Tip:** Es una buena práctica al finalizar el diseño, dejar solamente una biblioteca con todos los componentes utilizados dentro.

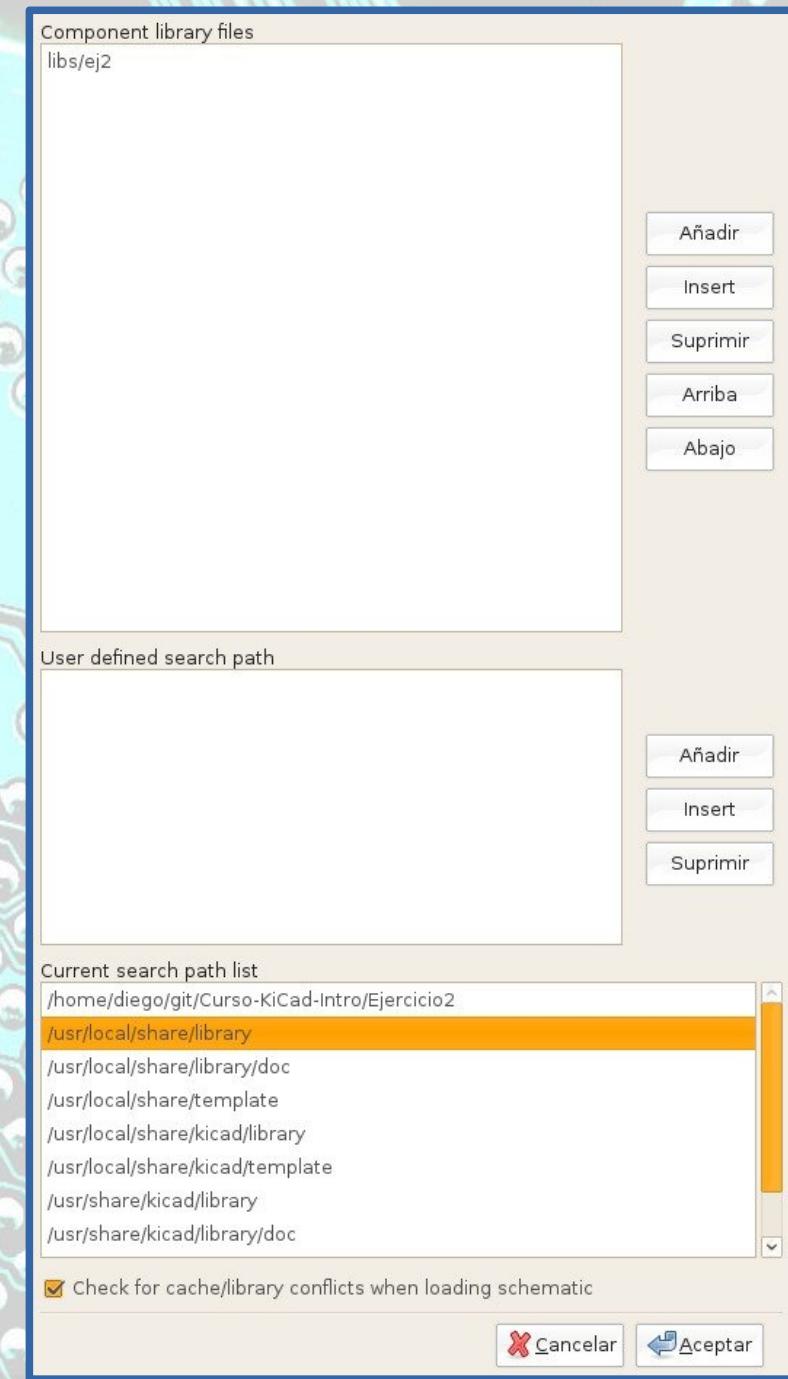
**Tip:** Al agregar una ruta (path) que está dentro de nuestro proyecto, es conveniente agregarla en forma relativa. Así no depende del lugar absoluto de trabajo.

Las bibliotecas provistas por KiCad pueden modificarse en el tiempo, pero nuestro diseño debe mantenerse inalterado.

Las bibliotecas de símbolos más usadas en KiCad:

- device (dispositivos de uso común)
- power (Etiquetas para alimentación)
- conn (Conectores)

**1) Dejar disponible solo la biblioteca provista y con ruta relativa.**



# Eschema - Jerarquía de hojas

## Diseño Jerárquico:

KiCad permite diseño jerárquico simple y complejo. Para el ejercicio utilizaremos una hoja principal y dos sub-hojas.

**Tip: Configurar siempre las páginas en tamaño A4 para permitir una correcta impresión del esquemático.**

PRINCIPAL

CONECTORyCAN

RS232

# Eschema - Jerarquía de hojas - Botones



Navegador de jerarquía

## Diseño Jerárquico: Flujo de trabajo

- 1) Agregar en la hoja principal las sub-hojas.
- 2) Editar cada sub-hoja creando los labels jerárquicos correspondientes.
- 3) Importar los pines jerárquicos en cada sub-hoja de la hoja principal.
- 4) Realizar las interconexiones entre hojas.

## Navegación entre hojas

- Para ir a cualquier hoja usar el navegador de la barra superior o el de la barra lateral derecha (funcionan diferente).



Subir/bajar jerarquía

### Tips

- Evitar usar labels globales ya que son difíciles de rastrear en el esquemático.
- Ubicar los labels jerárquicos en los bordes de las páginas para encontrarlos rápidamente.

2

- Crear label jerárquico en sub-hoja
- Crear nueva hoja jerárquica en hoja principal
- Importar pin jerárquico en la hoja jerárquica
- Crear pin jerárquico en hoja jerárquica

Este último botón no conviene usarlo, no es práctico!

# Ejercicio 2 - Jerarquía de hojas

Realizaremos una jerarquía de hojas con:

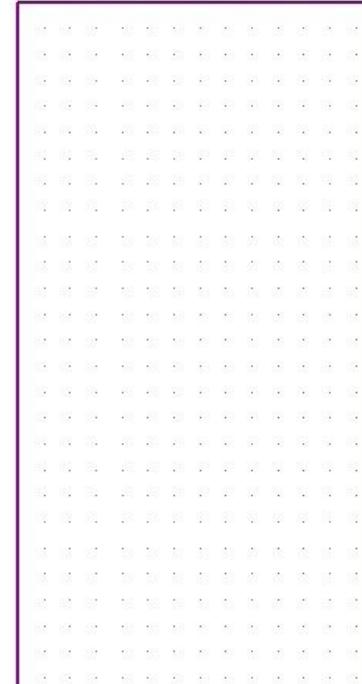
- Una página principal
- Una sub-hoja para el conector y el CAN.
- Una sub-hoja para el driver RS-232.

Sheet: Conector



File: Conecotor.sch

Sheet: RS-232



File: Rs232.sch

- 1) Crear dos hojas jerárquicas.
- 2) Crear los pines jerárquicos en la hoja del conector y CAN.
- 3) Importar los pines jerárquicos en la hoja principal.

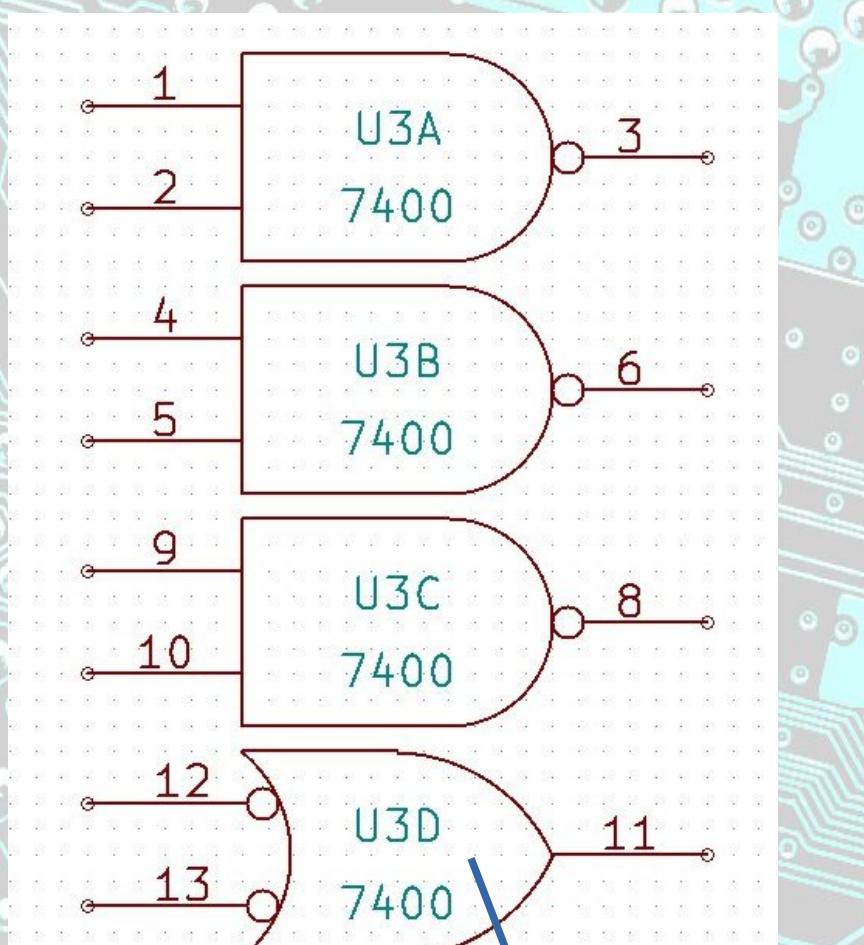
# Ejercicio 2

## Editor de símbolos

# Símbolos multipartes

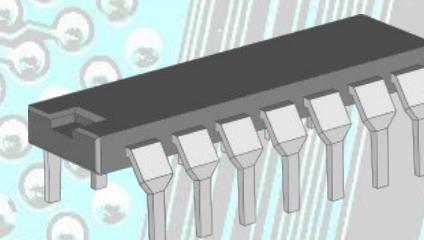
Para determinados componentes es de mucha utilidad la opción de símbolos multipartes.

Un símbolo multipartite utiliza varios símbolos (unidades) para representar un solo circuito integrado o componente.



## ¿Cuando se utiliza?

- Varios bloques funcionales similares.
- Muchos pines.
- Varias funciones diferentes.



El símbolo multipartite nos permite organizar mejor la hoja y también distribuir el componente entre diferentes hojas.

Algunos ejemplos de uso:

- Compuertas lógicas.
- Operacionales.
- FPGAs.
- Buffers

# Símbolos multipartes

Las compuertas 7400 anteriormente vistas se crearon con las siguientes opciones:

Opciones Description Alias Filtro Módulos

Alias List

- 74LS37
- 7400
- 74HCT00
- 74HC00

Añadir Borrar

Opciones Description Alias Filtro Módulos

Módulos (Footprints)

- 14DIP300\*
- SO14\*

Añadir Editar

Opciones Description Alias Filtro Módulos

General

- Has alternate symbol (DeMorgan)
- Show pin number
- Show pin name
- Place pin names inside

Number of Units (max allowed 64) Pin Name Position Offset

4 30

Define as power symbol

All units are not interchangeable

Cancelar Aceptar

Opciones Description Alias Filtro Módulos

Description

Quad nand2

Palabras clave

TTL nand2

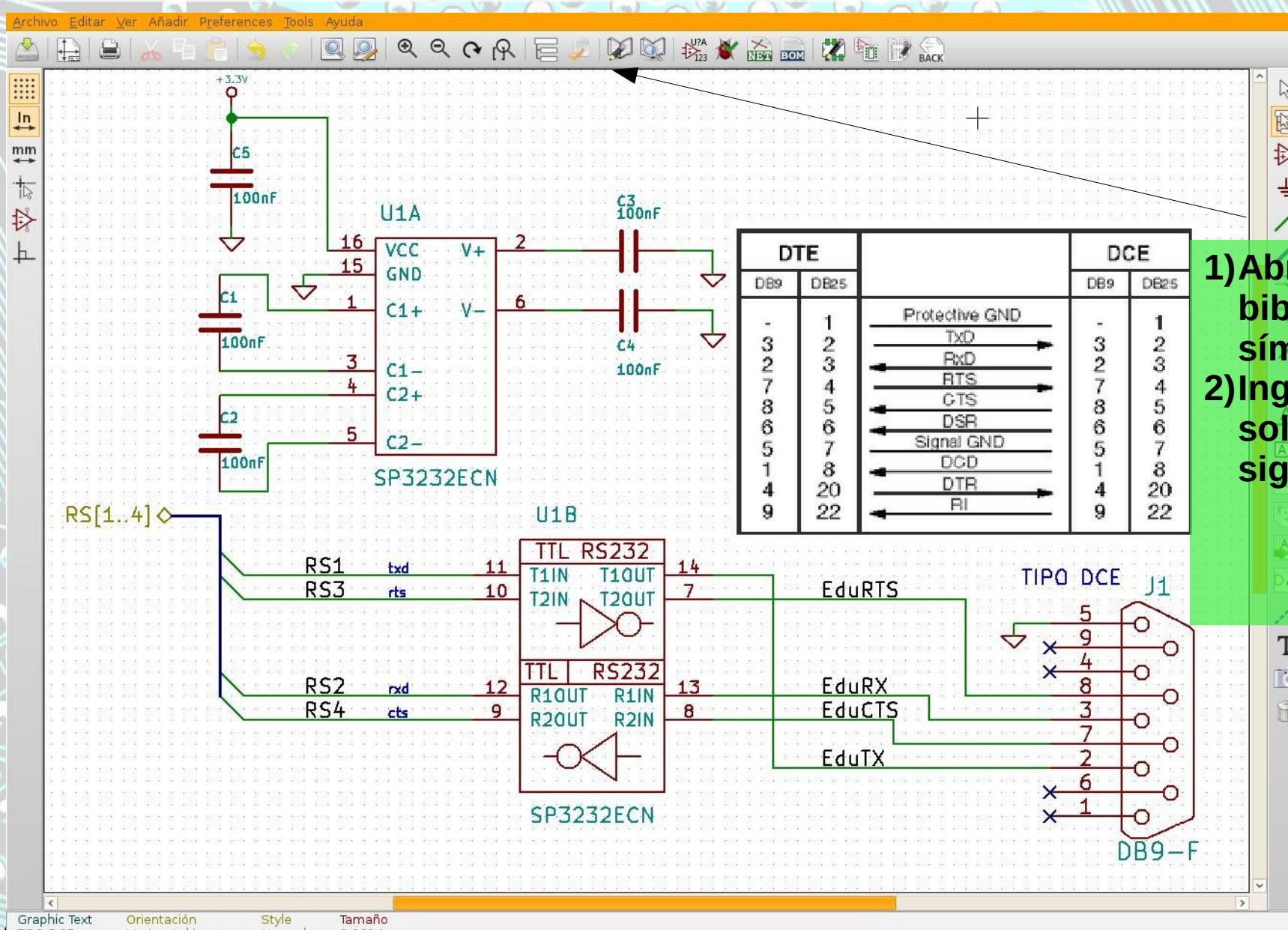
Documentation File Name

Copy Document from Parent Browse Files

Cancelar Aceptar

# Ejercicio 2 - Creación de un símbolo

En la hoja RS-232 realizaremos el siguiente circuito.  
El SP3232ECN lo crearemos desde cero para incursionar  
en la creación de símbolos.



- 1) Abrir el editor de bibliotecas de símbolos.
- 2) Ingresar los datos solicitados (ver siguiente slide).

# Ejercicio 2 - Creación de un símbolo

Actualmente el proyecto KiCad tiene pautas para la creación de símbolos. Estas pautas nos ayudan a ser organizados, consistentes y estar en sintonía con el software:  
<https://github.com/KiCad/kicad-library/wiki/Kicad-Library-Convention>

## General Rules

- Writing uses C-style naming with the first letter of each word being capitalized. Ex: **"Socket\_Strip\_Straight\_2x06"**
- Every acronym has all of its letters capitalized.
- Manufacturer name is capitalized as usual. Ex: NEC, Microchip.
- When dimensions are used in part name, they are in millimeters, decimal places separated by a dot, and unit is not mentioned. Ex: "C\_Rect\_L13\_W4\_P10"
- **Filename is the same as the part name.**
- The order of elements in names must be the same as the enumerations presented in this document.
- Reference fields are prefilled with the reference designator of the part (IEEE 315-1975).

## Symbol Library Names (.lib files)

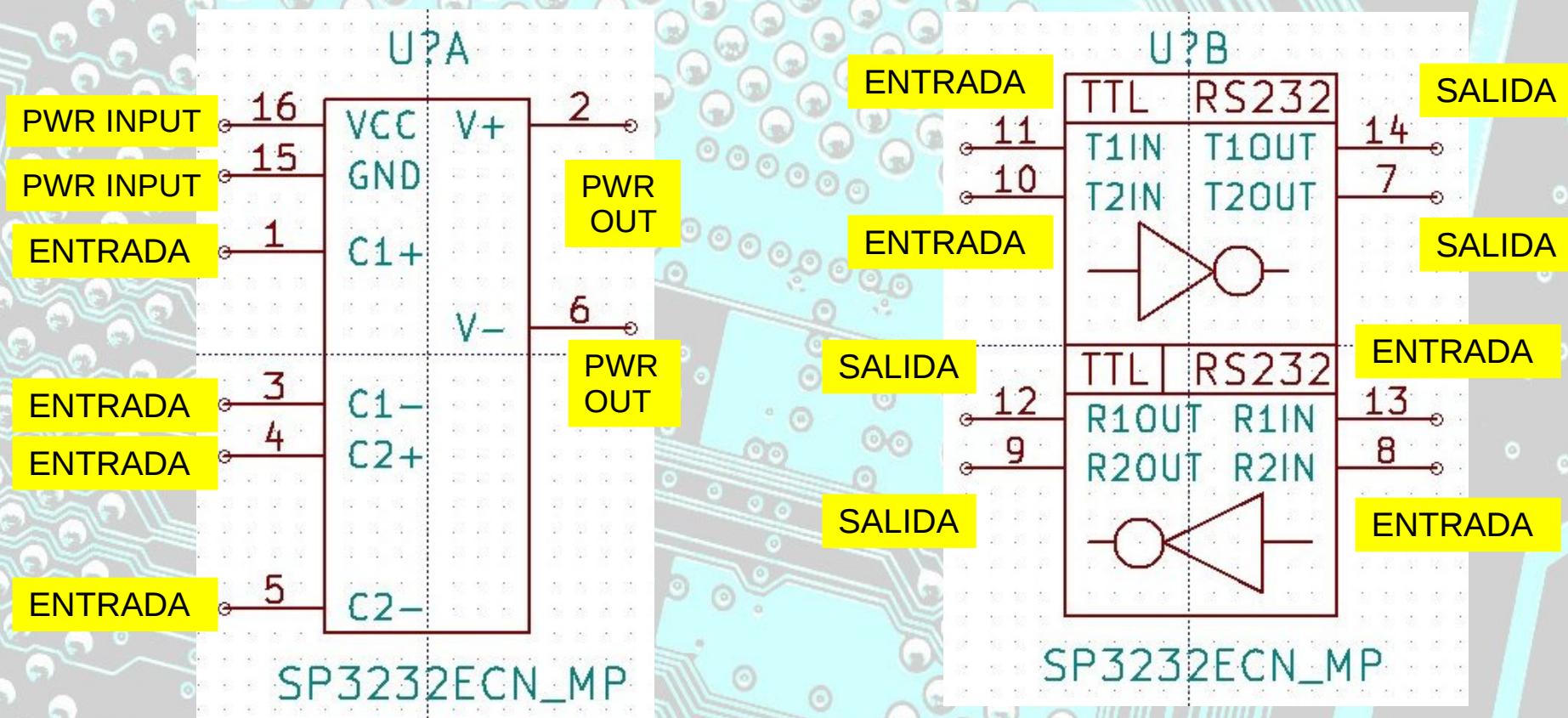
- Manufacturer.
- Category or family of parts. ex: "Capacitors", "Spartan6", etc.

## General Rules for Symbols

- **Using a 100mil grid**, pin ends and origin must lie on grid nodes (IEC-60617).
- For black-box symbols, pins have a length of 100mils. Large pin numbers can be accommodated by incrementing the width in steps of 50mils.
- **Origin is placed in the middle of symbol.**
- Black-box components **group pins logically**, for example by function set, and ports in counter-clockwise position.
- Whenever possible, **inputs are on the left and outputs are on the right**.
- **Field text uses a common size of 50mils.**
- **The Value field is prefilled with the object name.**
- **Description and keywords properties contain the relevant information.**

# Ejercicio - Creación de un símbolo multipartre irregular

Crearemos un símbolo multipartre para el SP3232. Dividiremos el componente en dos partes, una parte con todos los adaptadores de niveles TTL/RS232 y la otra parte con los capacitores y la alimentación.



- 1) Realizar el componente multipartre.
- 2) Guardarlo en el directorio de símbolos del ejercicio 2.
- 3) Ingresar ambas partes al esquemático y conectarlas.

Para cada pin de conexión  
hay que seleccionar una  
propiedad:

INPUT,  
PASSIVE,  
OPEN COLLECTOR,

OUTPUT,  
NO ESPECIFICADO,  
OPEN EMITTER,

BIDIRECCIONAL,  
POWER INPUT,  
NO CONNECTION

TRI-STATE,  
POWER OUTPUT,

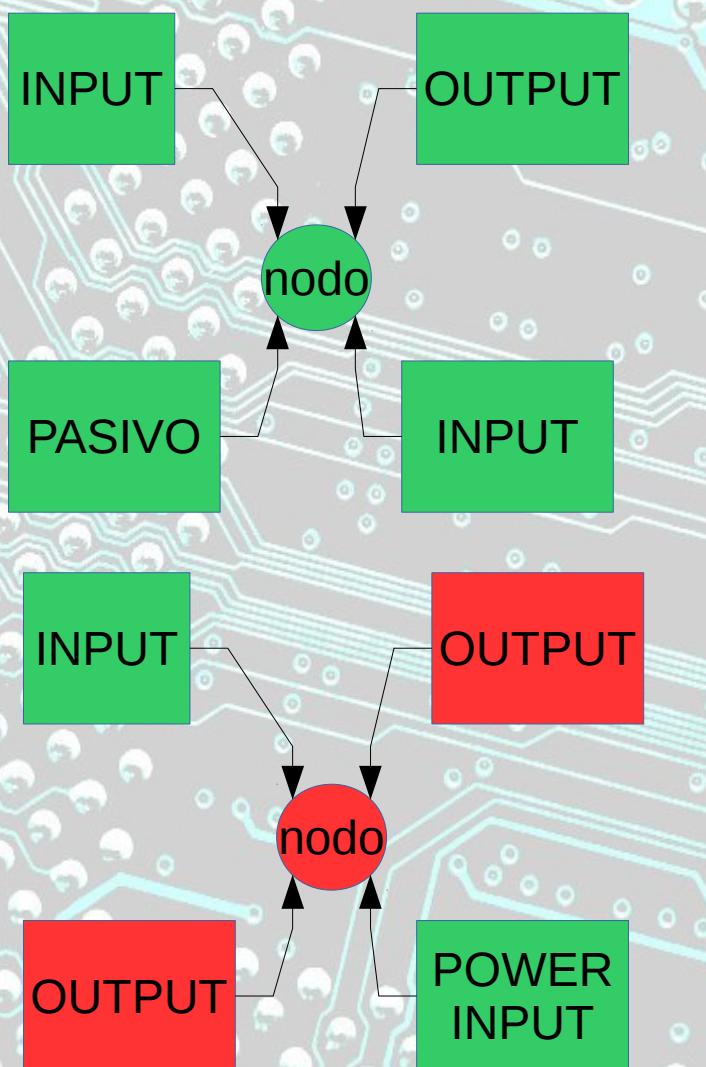
# Ejercicio 2

## Editor de esquemáticos

# Ejercicio 2 - ERC

Las propiedades de los pines son utilizadas por el ERC.

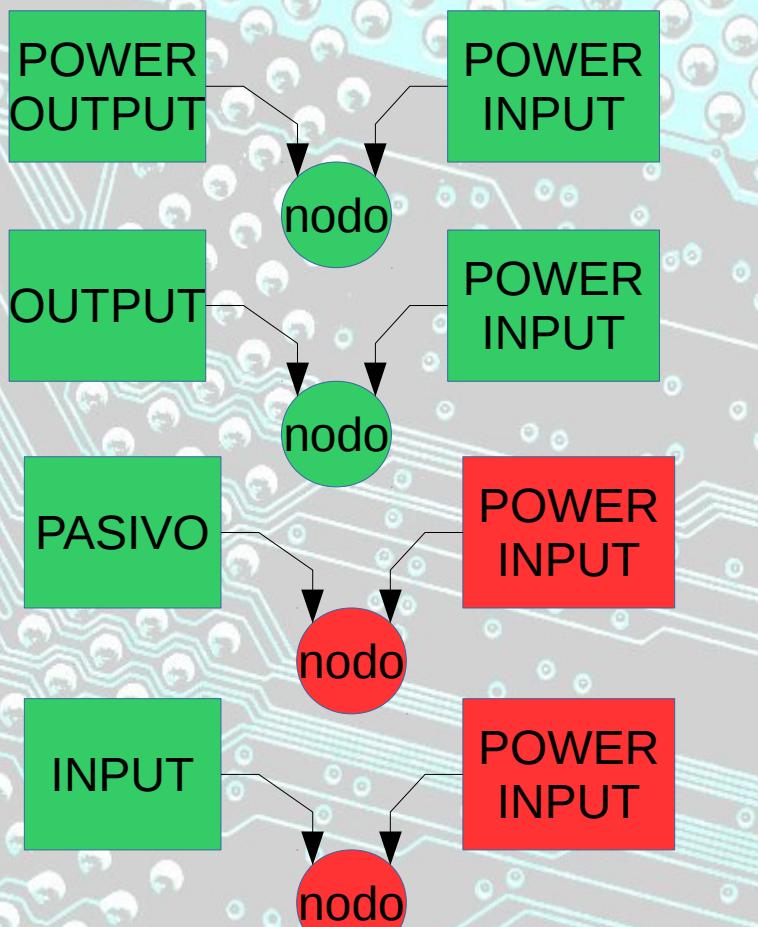
El ERC (Electrical Rules Checker) de KiCad NO SIMULA, solamente aplica esta matriz en cada nodo, considerando las propiedades de los pines conectados al mismo.



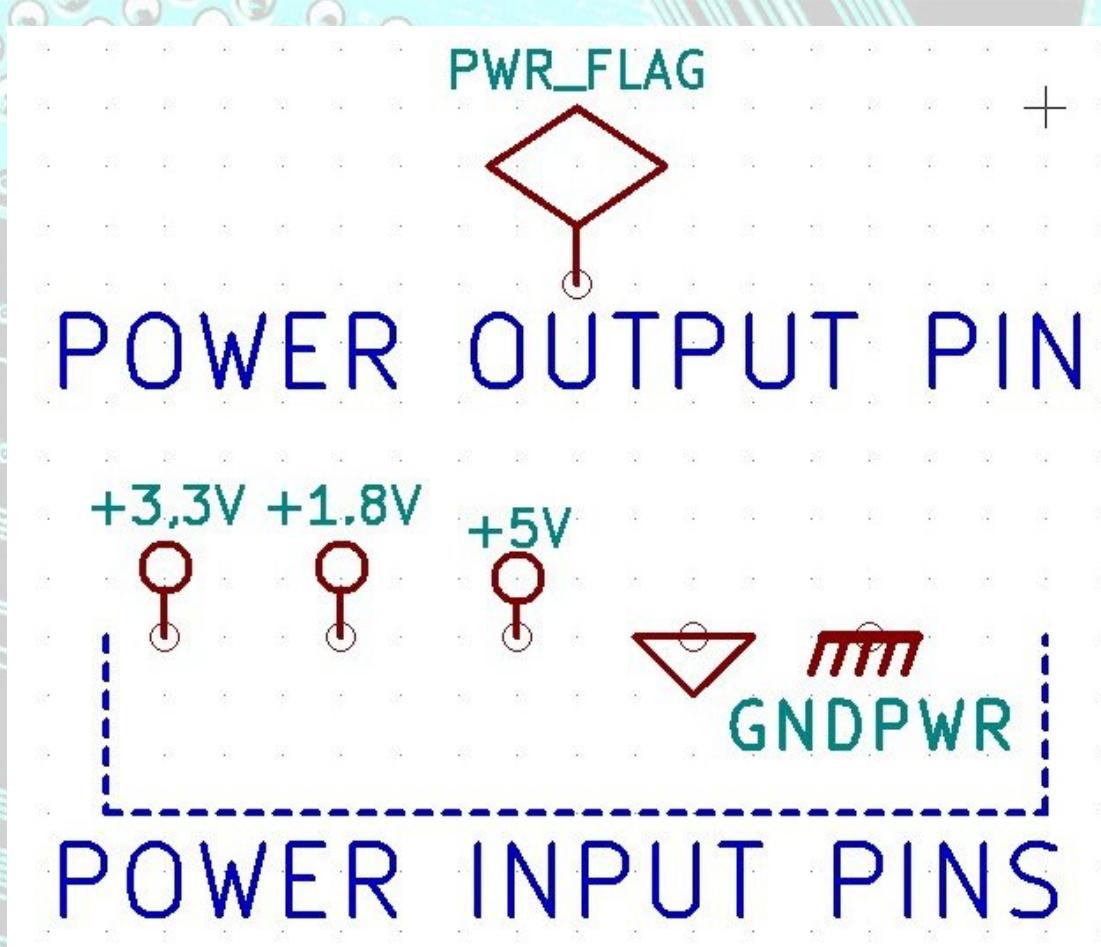
ERC Opciones		Defecto					
Input Pin.....	Output Pin.....	Input Pin	Output Pin	Bidirectional Pin	Tri-State Pin	Passive Pin	Unspecified Pin
Output Pin.....	E	Output Pin	E				
Bidirectional Pin..							
Tri-State Pin.....							
Passive Pin.....							
Unspecified Pin....							
Power Input Pin....							
Power Output Pin...							
Power Input Pin....							
Power Output Pin...							
Open Collector.....							
Open Emitter.....							
No Connection.....							

## Ejercicio 2 - ERC - Power

KiCad verifica que un pin de entrada de alimentación (power input) esté alimentado por alguno de los componentes del nodo.



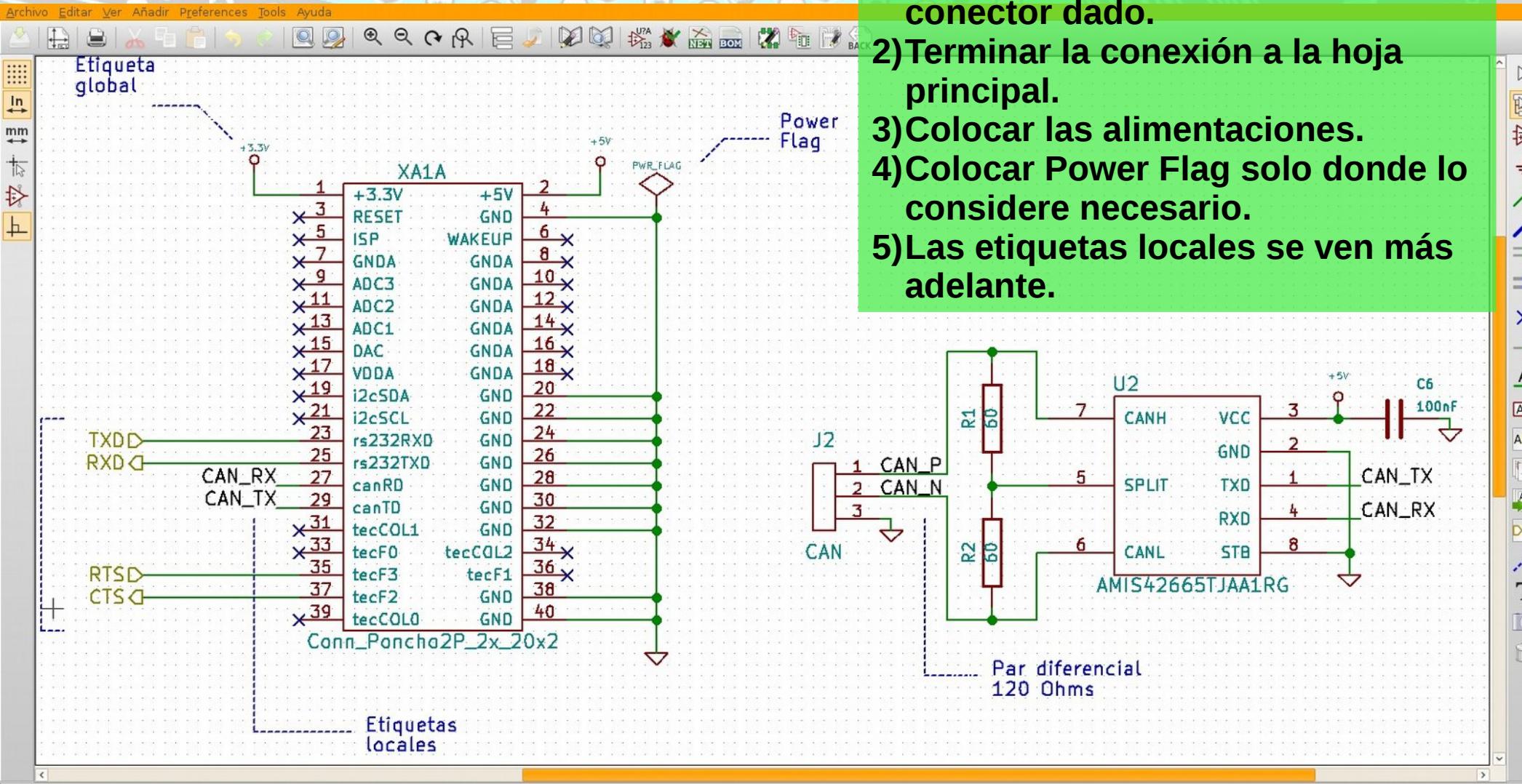
Además el ERC verifica que todos los componentes tengan una referencia válida, elementos no conectados, etc.



El POWER FLAG se utiliza para evitar el error de ERC cuando la alimentación proviene de un conector o viene de un componente con propiedad pasiva (fusible, diodo, resistor, etc.).

# Ejercicio 2 - Hoja conector

Completamos la hoja del conector, prestando atención a los power flags y las alimentaciones. El conector es un componente multipartite.



# Eschema - Etiqueta local

La etiqueta local permite realizar una conexión dentro de una hoja, sin que una línea realice todo el trayecto de conexión.

La etiqueta debe aplicarse sobre un pin o sobre un tramo de cable.

El punto de conexión es un cuadrado debajo de la primera letra de la etiqueta, y debe estar en contacto con el pin o con el tramo de cable.

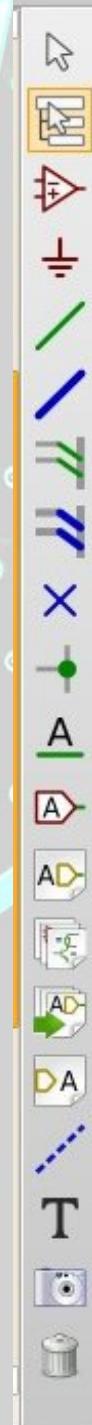
**TIP:** Las etiquetas pueden hacer más claro un circuito esquemático al eliminar líneas que cruzan por la hoja. Pero su abuso conduce a esquemáticos sin interconexiones visibles que dificultan la interpretación (como leer un archivo netlist).

Las etiquetas también sirven para definir buses.

RX

Etiqueta

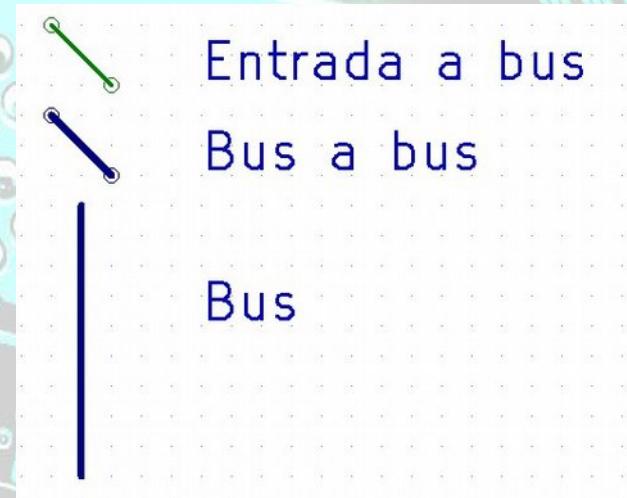
Etiqueta local



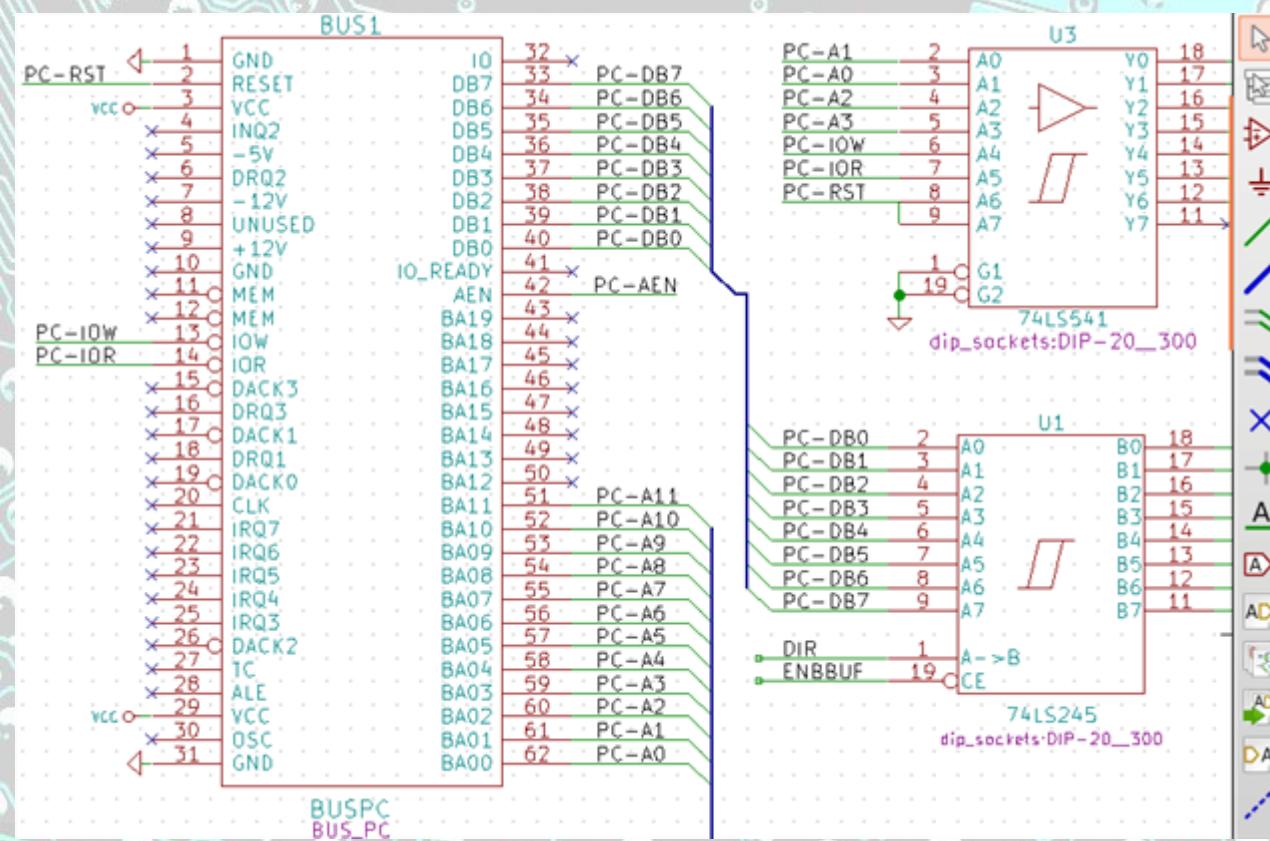
# Eschema - Buses

Los símbolos de entrada a bus y de bus a bus son solo dibujos para prolijidad en la hoja. Son estéticos y no cumplen función ni interconectan elementos.

La línea de bus también es estética y no cumple función de interconexión, salvo que se utilice una etiqueta o un puerto con nomenclatura de bus (siguiente slide),



Línea de bus  
Entrada a bus  
Entrada de bus a bus



En el circuito de la izquierda (PC-DBx) la interconexión real se hace con etiquetas locales. Las entradas de bus y los tramos de bus podrían borrarse y las interconexiones seguirían correctas.

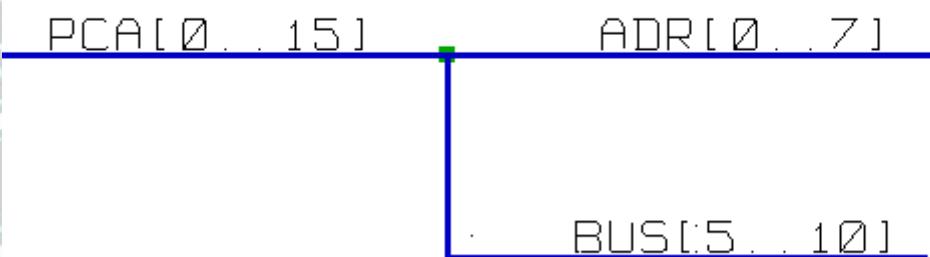
Etiqueta local

**Tip:** Para colocar las etiquetas de un BUS, colocar solo la primera y luego la tecla **[INS]** (Repeat las item). En las opciones de edición se configuran los parámetros de repetición.

# Eschema - Buses

## Nomenclatura de BUS

- Se utiliza un nombre de base (prefijo) y entre corchetes el rango de señales del bus [n..N], siendo el primer número de señal y N el último.
- Esta nomenclatura se puede utilizar en labels de conexión jerárquicos o sobre líneas de buses.
- Para mezclar buses, kicad simplemente conecta números iguales entre sí.
- Para referenciar cada señal del bus se utiliza la herramienta de etiqueta local.



conn[0..15]

con[0..15]

### CONEXIONES

PCA0-ADR0  
PCA1-ADR1  
PCA2-ADR2  
PCA3-ADR3  
PCA4-ADR4  
PCA5-ADR5-BUS5  
PCA6-ADR6-BUS6  
PCA7-ADR7-BUS7  
PCA8-BUS8  
PCA9-BUS9  
PCA10-BUS10  
PCA11  
PCA12  
PCA13  
PCA14  
PCA15

Línea de bus

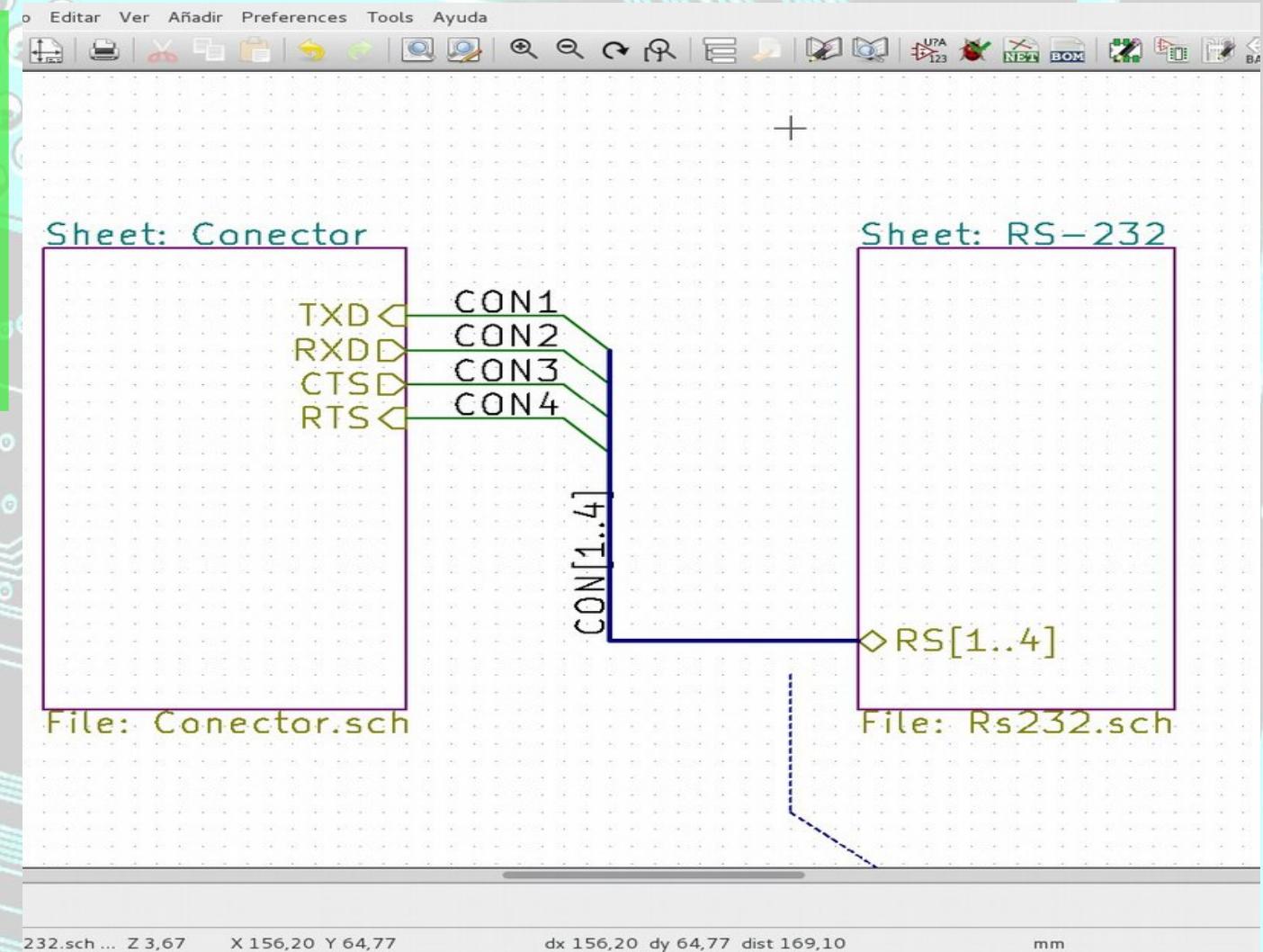
Etiqueta local

label en sub-hoja

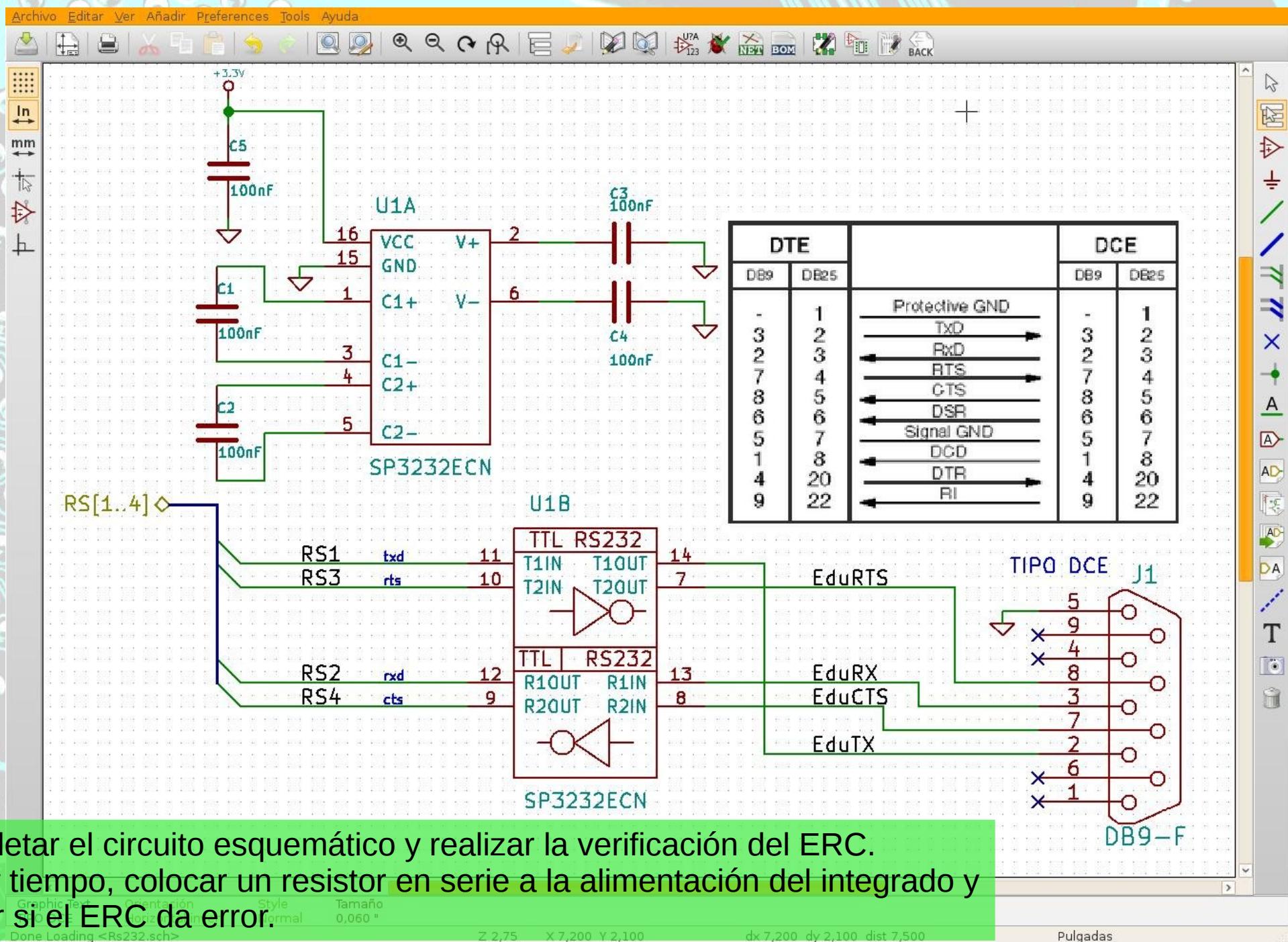
Label en hoja

# Eschema - Jerarquía de hojas y buses - Ejercicio 2

- 1) Utilizar un bus para interconectar ambas hojas.
- 2) En la hoja RS-232 ingresar con el bus para luego descomponerlo dentro de la sub-hoja.



# Ejercicio 2 - Interconexiones y verificación general



# Eschema - Campos de Información

## Campos de información: Por qué se usan?

- Correcta identificación de cada componente desde el momento que el desarrollador lo agrega al esquemático.
- Uniformidad de la descripción de todos los componentes.
- Obtención de una lista de componentes completa y ordenada para realizar la compra de materiales.

## Campos de información: Cómo se usan?

- 1) Acordar con todos los desarrolladores los campos que se utilizarán y cómo se completarán.
- 2) Chequear que todos tengan los campos escritos exactamente igual, no es lo mismo NRO. DE PARTE que NÚMERO DE PARTE!
- 3) Completar los campos a medida que se van agregando los componentes al esquemático, no dejar para el final.
- 4) No cambiar ni agregar campos durante el transcurso del proyecto para no generar conflictos.

Schematic Editor Options		
General Options		Template Field Names
Field Name	Default Value	Visible
Descripción		Hidden
Fabricante		Hidden
Nro. parte		Hidden
Path datasheet		Hidden
Digikey/Mouser		Hidden
Footprint estandar		Hidden

Preferences → Schematic Editor Options → Template Field Names

Field Settings

Name: Footprint estandar

Default Value:

Visible

Add

Component Properties

Fields

Name	Value
Reference	R206
Value	10K
Footprint	r_0603
Datasheet	~
Descripción	RES 4.7K OHM 1/10W
Fabricante	Yageo
Nro. parte	RC0603JR-0710KL
Path datasheet	~
Digikey/Mouser	311-10KGRCT-ND
Footprint estandar	0603

Horiz. Justify:

- Left
- Center
- Right

Vert. Justify:

- Bottom
- Center
- Top

Visibility:

- Show
- Rotate

Style:

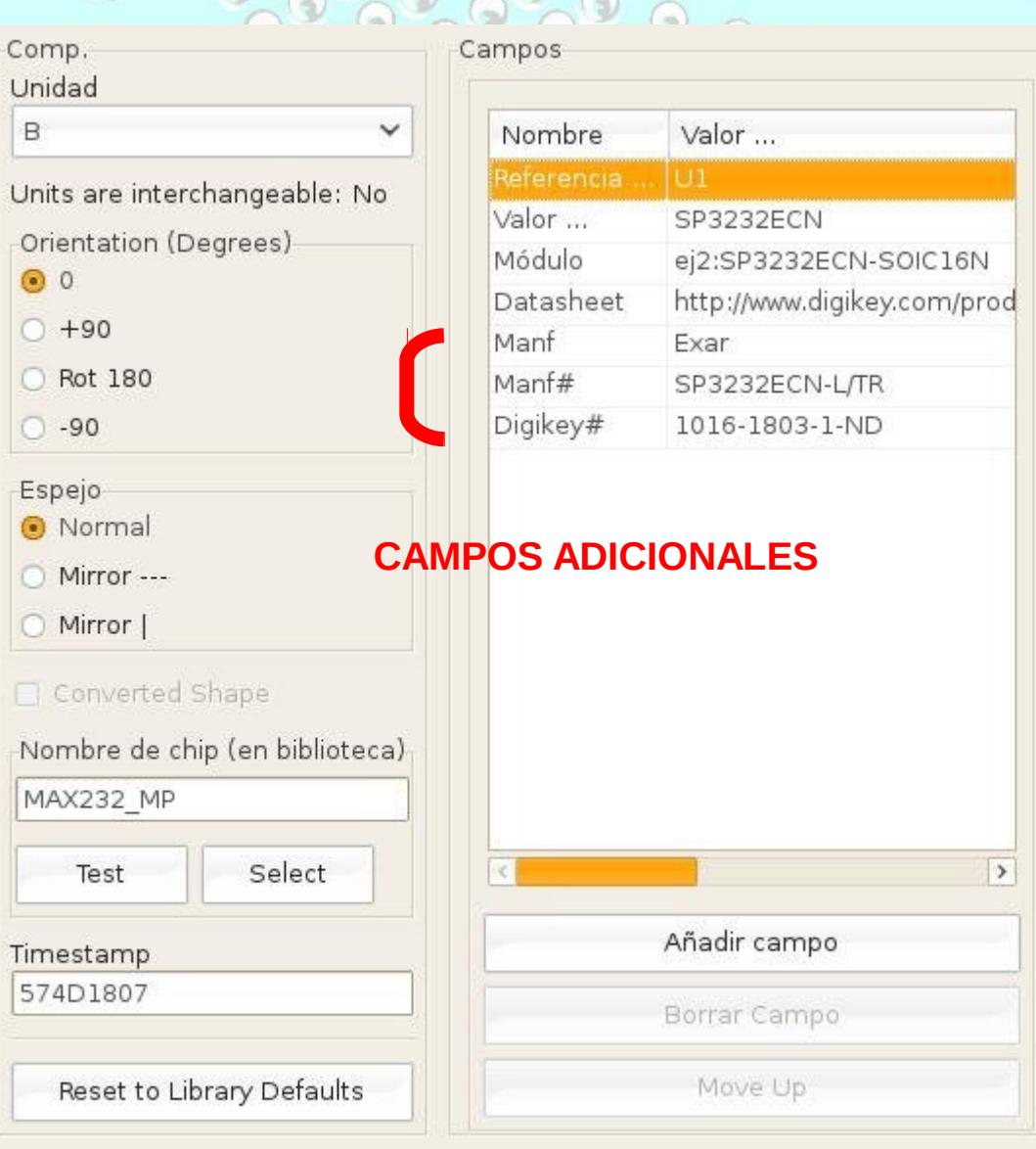
- Normal
- Italic
- Bold
- Bold Italic

Field Name: Reference

Field Value: R206

**Tip: Es posible editar campos desde los archivos de texto!**

# Eschema - Campos de información



En un diseño profesional es necesario colocar información del fabricante, Nro de parte, código del distribuidor, hoja de datos, etc.

También se podrían agregar campos como precio estimado, rango de temperatura, precios locales, stock local y varios más.

**Tip:** Utilizar los nombres definidos por KiCost para luego calcular automáticamente el presupuesto y el pedido.

Field Name  
Referencia ...

Field Value  
U1

Tamaño 0,060

PosX -0,150

PosY 0,600

**Campos sugeridos:**  
**Desc:** Descripción larga.  
**Manf:** Fabricante.  
**Manf#:** Código de fabricante.  
**Digikey#:** Código Digikey.  
**Mouser#:** Código Mouser.  
**Newark#:** Código Newark.

## Ejercicio 2 - Campos de información

- 1) Crear los campos de información sugeridos.
- 2) Completar el campo Digikey#.
- 3) Completar de tarea los campos Manf y Manf#.

Reference(s)	Value	LibPart	Footprint	Digikey#
C1, C2, C3, C4, C5, C6	100nF	ej2:C	ej2:C_0603_HandSoldering	311-1179-1-ND
J1	DB9-F	ej2:DB9	ej2:DB9_F_TH	S9554-ND
J2	CAN	ej2:CONN_3	ej2:CON_PALETA_3	A19470-ND
R1, R2	60	ej2:R	ej2:R_1206_HandSoldering	Y149660R0000A0R-ND
U1	SP3232ECN	ej2:SP3232	ej2:SP3232ECN-SOIC16N	1016-1803-1-ND
U2	AMIS42665TJAA1RG	ej2:AMIS42665TJAA1RG	ej2:SOIC-8	766-1006-1-ND
XA1	Conn_Poncho2P_2x_20x2	ej2:Conn_Poncho2P_2x_20x2	ej2:Conn_Poncho_Derecha	952-1387-ND

# BOM - Listado de materiales

Actualmente kicad genera un xml. Luego utiliza scripts python para generar distintos tipos de lista de materiales.

**En el directorio tools copiarse los siguientes archivos:**

kicad\_netlist\_reader.py (1)  
bom\_csv\_grouped\_by\_value.py (2)

(1) Script base que parsea el xml.  
(2) Script que genera el BOM deseado (llama a 1).

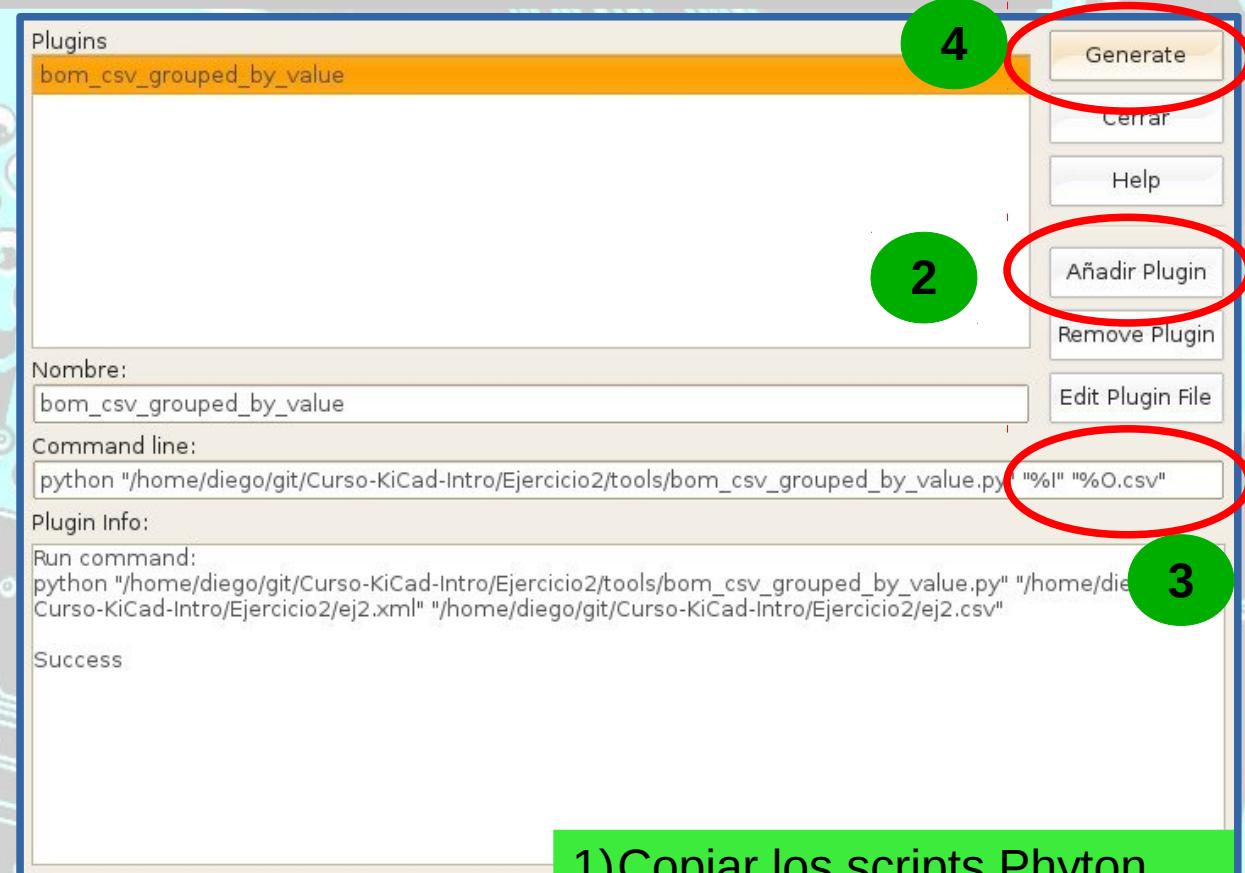
Luego añadir el plugin deseado.

Se puede editar la línea de comandos que usará Kicad. Por ejemplo cambiar a "%O.csv"

Generar el BOM y abrirlo lo una planilla de cálculo.

**TIP:** Hay más scripts en:

<https://github.com/KiCad/kicad-source-mirror/tree/master/scripts/bom-in-python>



Kicad generará un bytecode de Python:  
kicad\_netlist\_reader.pyc

**Alerta:** Por el momento KiCad guarda las rutas absolutas de estos scripts y fuera del proyecto.

- 1)Copiar los scripts Phyton.
- 2)Agregar el plugin.
- 3)Editar la línea de comandos.
- 4)Generar el BOM.
- 5)Abrirlo con un software de planilla de cálculo.

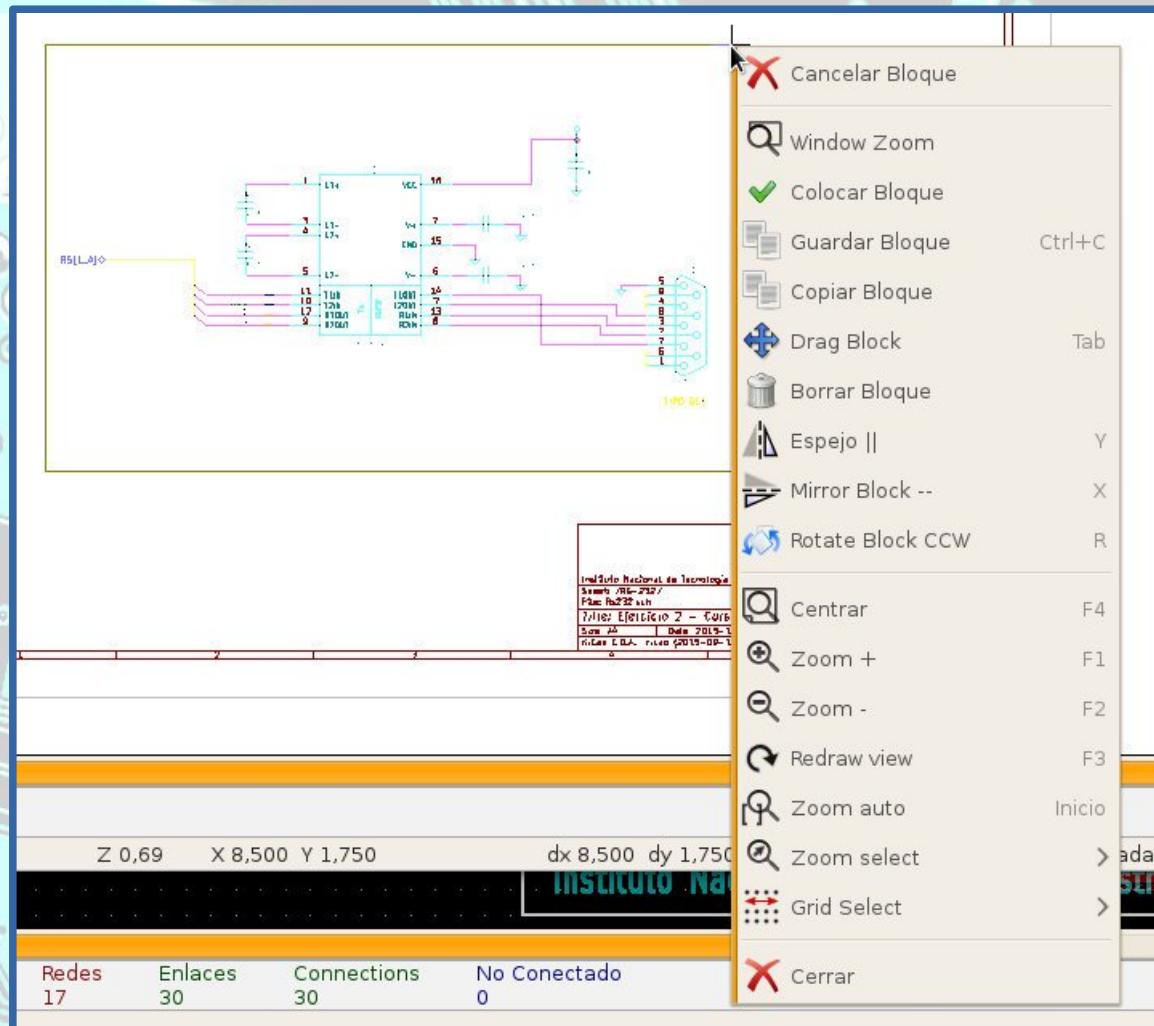
# Eschema - Operaciones de bloques

En el esquemático podemos realizar algunas operaciones de bloques como mover, copiar, espejar o borrar.

Para realizar las operaciones de bloques:

- 1)Presionar el botón izquierdo de mouse en la primer esquina del bloque.
- 2)Manteniendo presionado, mover el mouse hasta la otra esquina del bloque a definir.
- 3)Soltar el botón izquierdo y presionar el derecho para desplegar el menú de operaciones de bloques.
- 4)Si no se presiona el botón derecho, se aplica la operación mover bloque.

- 1)Probar operaciones de bloques, usando el “deshacer” luego de cada una.
- 2)Intentar copiar un bloque de una hoja a la otra. Ayuda: usar “Guardar bloque”.



**Tip:** En KiCad todavía no se puede confiar “ciegamente” en la función deshacer. Es una característica agregada recientemente y algunas pocas operaciones en Eeschema y Pcbnew no tienen deshacer.

# Ejercicio 2

## Asociación de footprints

## Ejercicio 2 - Asociación de footprints

Con las bibliotecas de footprints dadas para el ejercicio realizar la asociación de todos los footprints. Ver tabla anterior.

- 1)Copiar los footprints dados al directorio ej2.pretty
- 2)Usando el Wizard ingresar el directorio como en el ejercicio 1 (solo para este proyecto).
- 3)Usando Cvpcb o mediante el campo “Módulo” de cada símbolo realizar la asociación de footprints.
- 4)Generar el netlist.
- 5)Grabar el esquemático.



# Sobre la asociación de footprints

Se debe tener especial cuidado al asociar un símbolo con un footprint.  
La problemática es la siguiente:

1) Existe una gran cantidad de símbolos genéricos que se pueden utilizar para una gran variedad de componentes diferentes (como por ejemplo el símbolo de un transistor NPN).

2) Existe una gran cantidad de footprints estandarizados como por ejemplo TO-92, TO-220, TO-3, TO-18, SOT-23. El estándar del encapsulado define la forma y las dimensiones, pero no la función de cada pin o el tipo de dispositivo.

3) Lo que termina definiendo la relación entre la funcionalidad del pin y su correspondiente pin en el encapsulado es la hoja de datos.



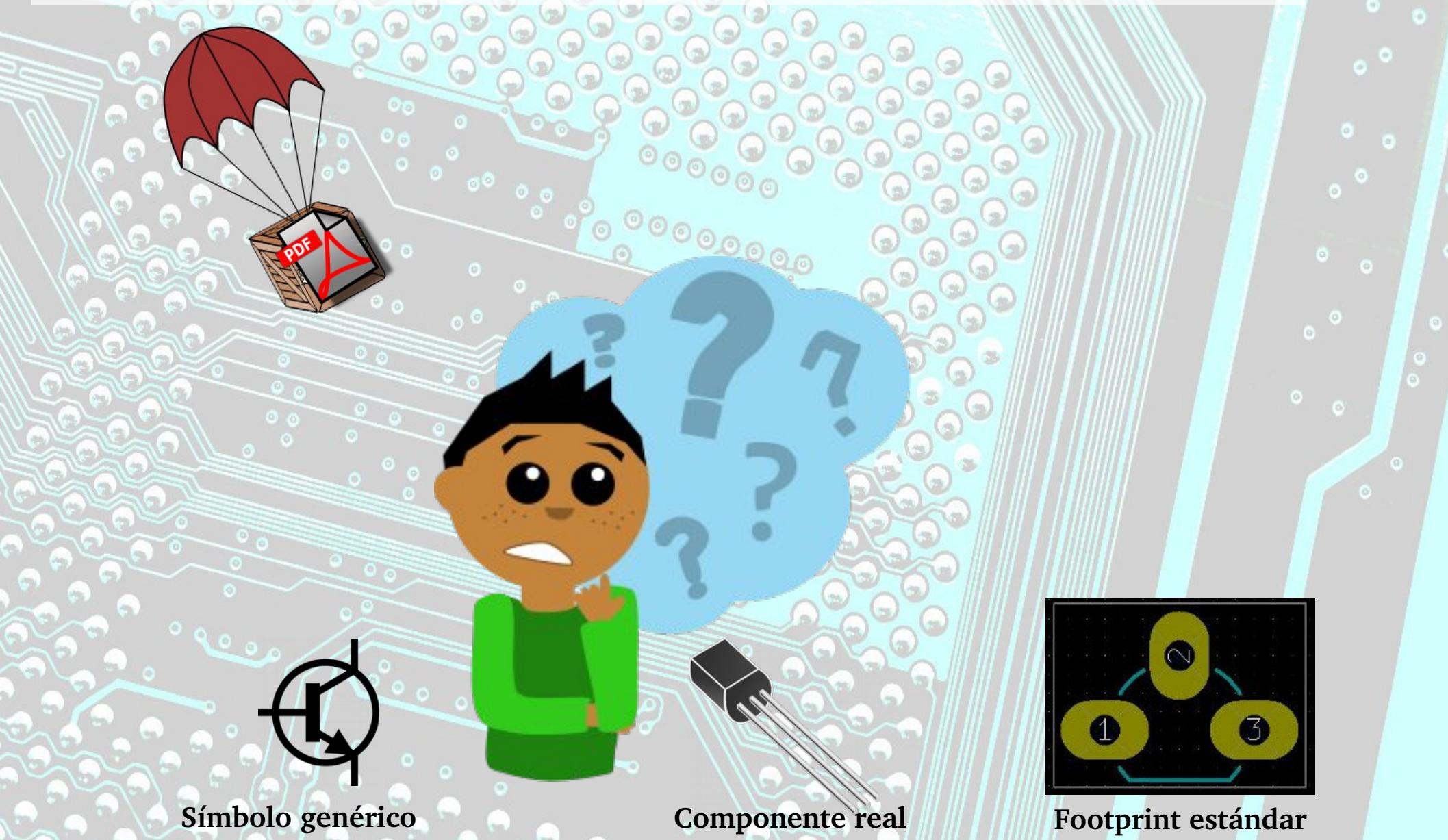
<https://commons.wikimedia.org/wiki/File:Transbauformen.jpg>



Se presenta al transistor NPN como ejemplo. La problemática es general de las bibliotecas, que evitan considerar la infinidad de componentes electrónicos existentes cuando no es necesario.

# Sobre la asociación de footprints

Es necesario consultar la hoja de datos, analizar y realizar los ajustes necesarios:



Símbolo genérico

Componente real

Footprint estándar

# Sobre la asociación de footprints

El ejemplo del transistor NPN en KiCad.  
Supongamos tenemos un BC548:

1) El símbolo existente está en la biblioteca "devices" y se llama NPN. Solo se debe editar el campo valor para reflejar nuestro modelo particular de transistor.

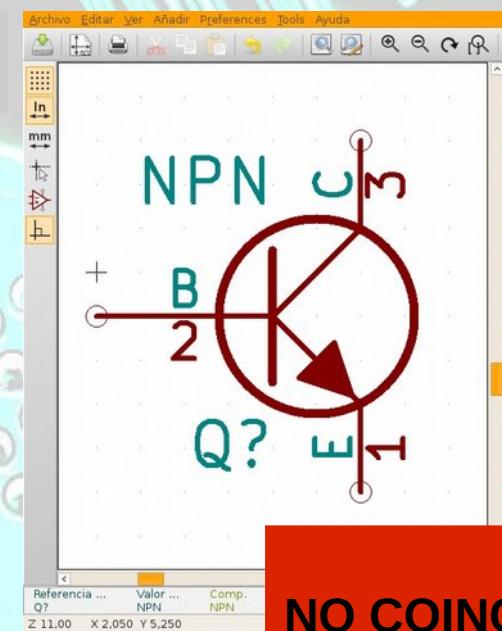
2) Seleccionamos de la biblioteca de footprints (TO\_SOT-Packages-THT) alguno de los modelos TO-92 existentes.

3) La asociación que realiza KiCad es simple:

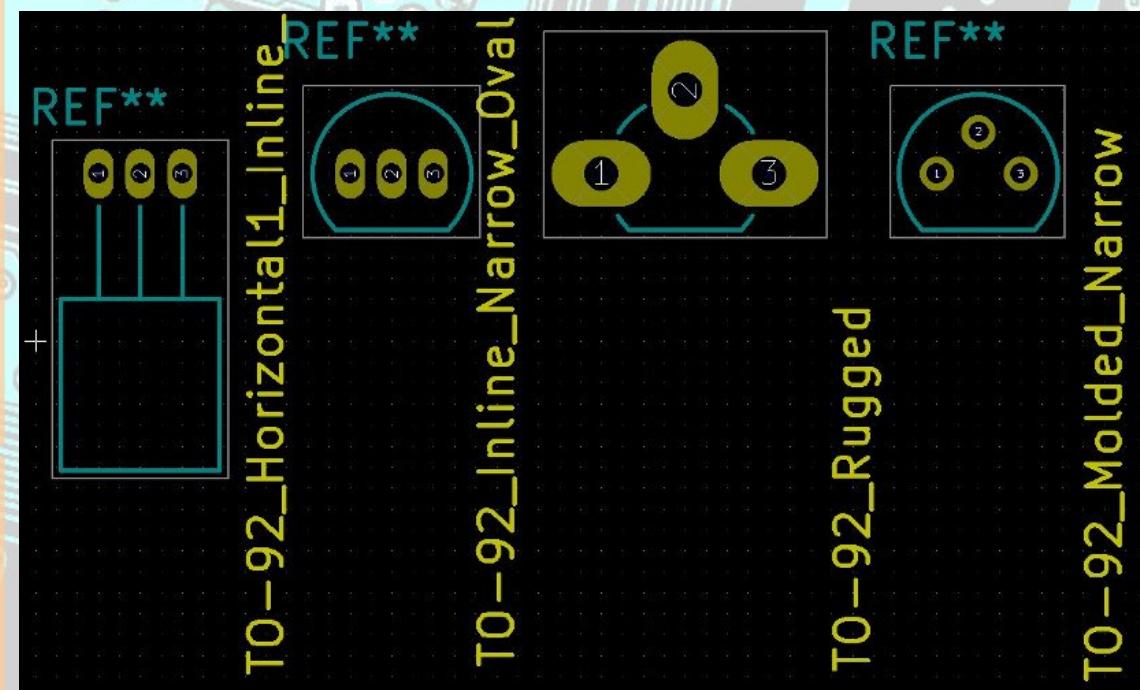
- 1- emisor - 1 footprint
- 2- base - 2 footprint
- 3- colector – 3 footprint

Verificamos si coincide la numeración en el símbolo con la del footprint, mirando la hoja de datos.

4) Si no coincide, la convención definida en KiCad es la de editar el símbolo esquemático. Luego generar el netlist y pasar al PCB.



NO COINCIDE



# Ejercicio 2

## Editor de PCB

# KiCad Graphics Abstraction Layer (GAL)

La GAL es una capa de abstracción de gráficos que se incorporó a KiCad a partir de mediados de 2013.

Previamente a la existencia de la GAL cada herramienta debía considerar los aspectos de dibujo y manejo de gráficos (dibujar y borrar líneas por ejemplo).

El uso de la GAL hace el código de cada herramienta más simple y desacopla funcionalidades con métodos gráficos.

Para utilizar la GAL fue necesario reescribir todas las herramientas de KiCad para que la utilicen.

Mucha de las nuevas herramientas ya no soportan el modo tradicional (sin GAL).

Modo legacy

Modo GAL

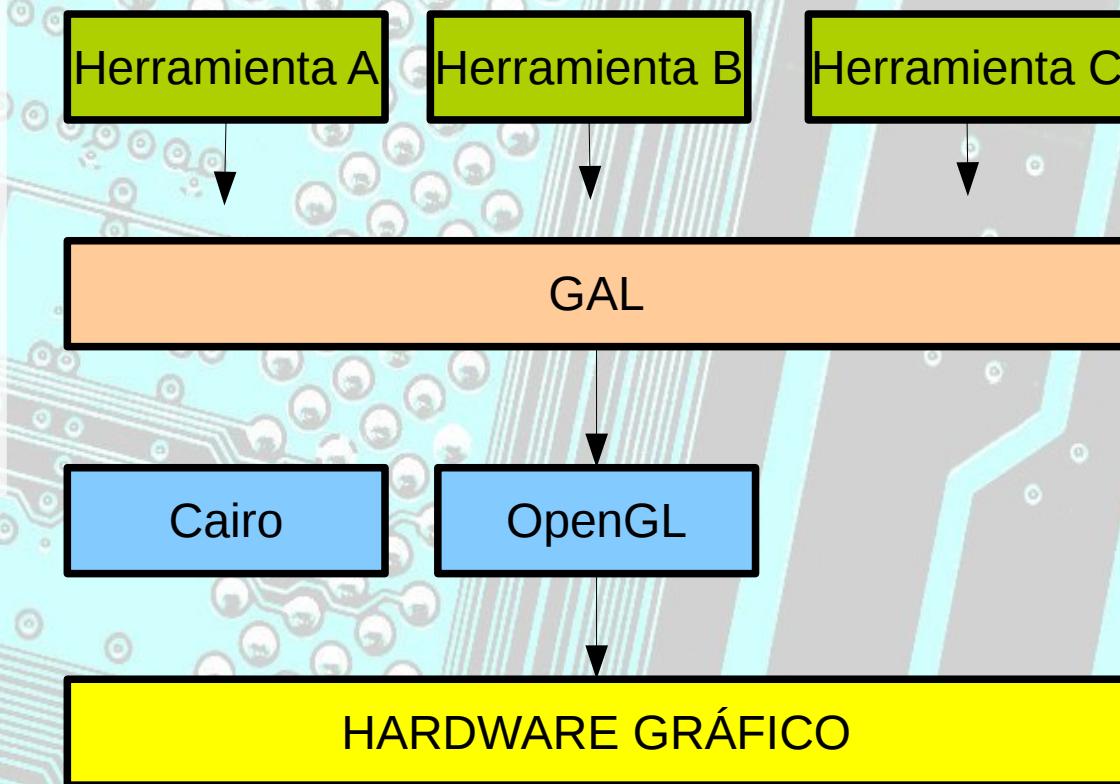
Herramienta no portada

Herramienta portada

Herramienta portada

Herramienta nueva

*En las versiones actuales de KiCad conviven herramientas no portadas, portadas y nuevas. Estas últimas solo funcionan con la GAL.*

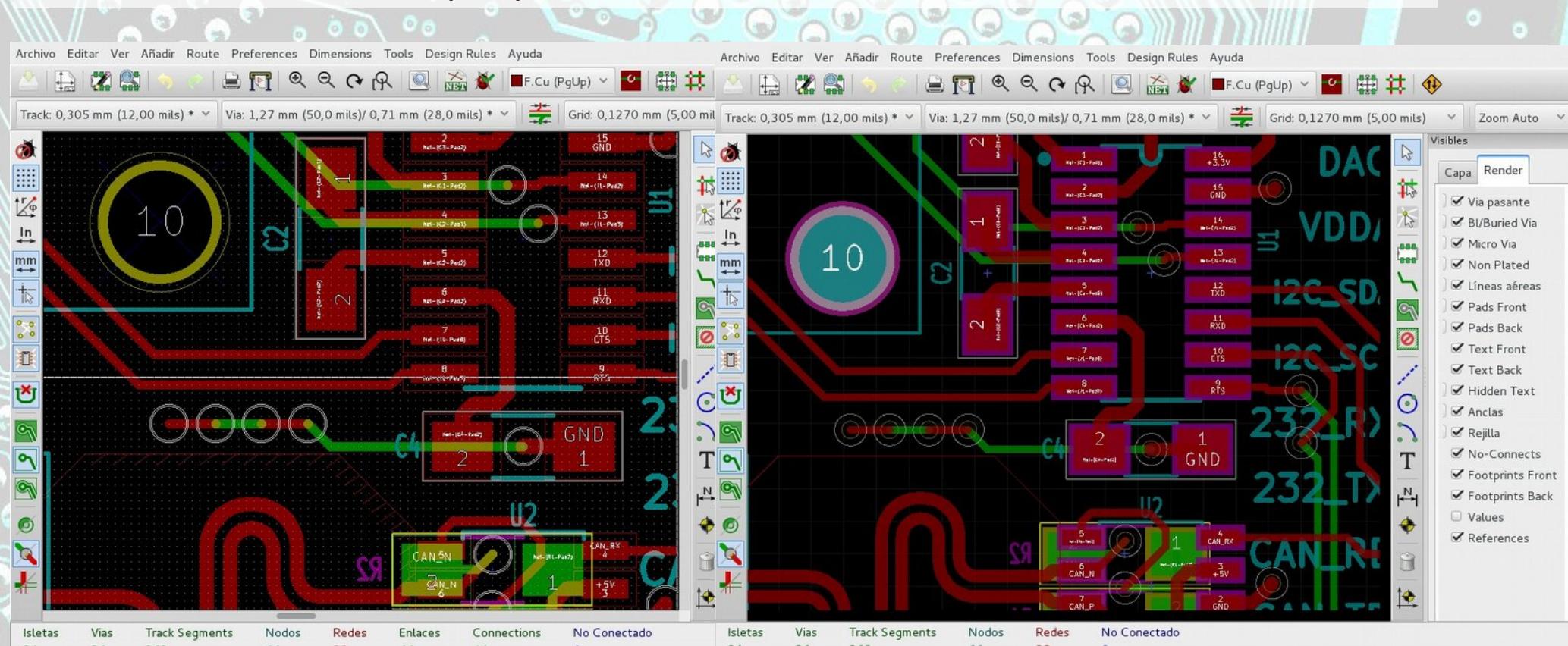


# KiCad Graphics Abstraction Layer (GAL)

La GAL esta funcionando actualmente solo para Pcbnew y el editor de módulo.

Para cambiar la visualización entramos a la opción del menú “Ver” y luego:

- Switch canvas to default (F9).
- Switch canvas to OpenGL (F11).
- Switch canvas to Cairo (F12).



Modo default

- 1) Cambiar al modo OpenGL.
- 2) Mover tracks y footprints.

Modo OpenGL

# Pcbnew: Modos de visualización

The image shows a screenshot of the KiCad PCB Editor interface. On the left, there is a vertical toolbar with various icons for editing tools. A green box highlights three specific icons: a key icon (Mode 1), a magnifying glass icon (Mode 2), and a red X icon (Mode 3). To the right of the toolbar, there is a large area showing a printed circuit board (PCB) with various components and traces. Overlaid on this are several text annotations and diagrams illustrating different visualization modes:

- Cuando se trabaja en un PCB multicapa, es necesario facilitar la visualización para tener claridad al rutear.** (When working on a multi-layer PCB, it is necessary to facilitate visualization to have clarity when routing.)
- Lo ideal es tener siempre todas las capas de cobre habilitadas y usar el modo de alto contraste para que sólo una capa se vea en color y las demás en gris.** (It is ideal to always have all copper layers enabled and use the high contrast mode so that only one layer is seen in color and the others in gray.)
- Qué opciones nos da KiCad?** (What options does KiCad give us?)
- Modo de alto contraste Usar tecla 'H'** (High contrast mode Use 'H' key)  
Diagram: A black background showing red traces and pads. Labels include GND, 55, 53, SD\_D2, SD\_D0, SD\_PWR, SD\_CMD, BUS, and 43.
- Áreas de cobre ocultas** (Hidden copper areas)  
Diagram: A black background showing blue traces and pads. Labels include 55, SD\_D2, 53, SD\_D0, SD\_PWR, SD\_CMD, 47, SD\_PWR, 45, SD\_CMD, 43, SD\_CD, and SD\_PWR.
- Modo Outline De pistas, vías y pads (sólo bordes)** (Outline mode Traces, vias and pads (only borders))  
Diagram: A black background showing only the outer borders of the traces and pads. Labels include 55, SD\_D2, 53, SD\_D0, SD\_PWR, SD\_CMD, 47, SD\_PWR, 45, SD\_CMD, 43, SD\_CD, and SD\_PWR.
- También se pueden “vaciar” los planos con Ctrl+B** (Also, the planes can be “emptied” with Ctrl+B)

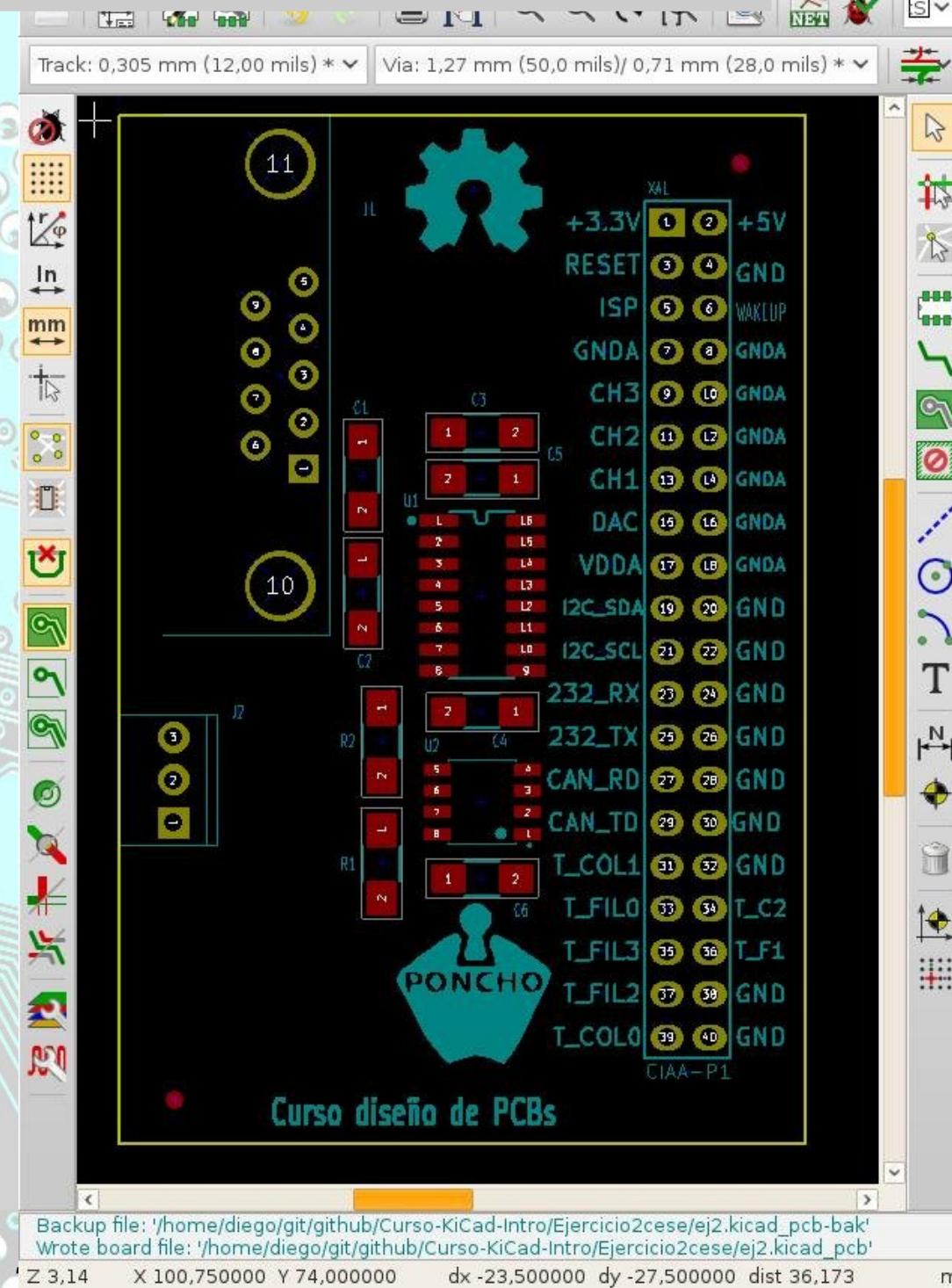
At the bottom right, there is a Creative Commons BY-SA 4.0 license logo.

# Ejercicio 2 - Importar netlist

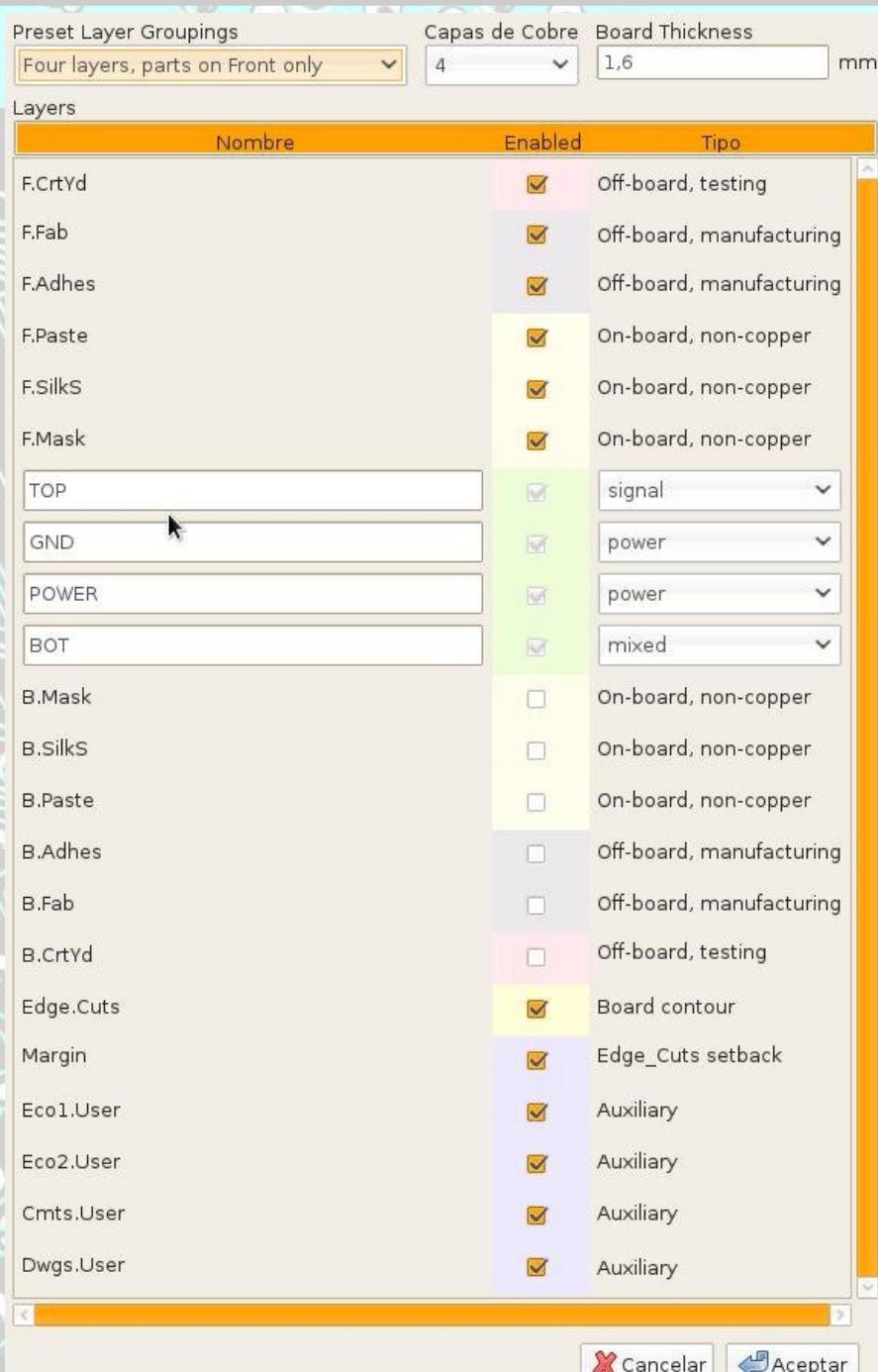
In Rules Ayuda

En el PCBnew ingresamos el netlist, separamos los componentes, dibujamos el borde de impreso y realizamos un placement rápido.

- 1)Ingresar el netlist al pcbnew.
- 2)Separar componentes automáticamente.
- 3)Dibujar borde de impreso (Usar plantilla EDU-CIAA).
- 4)Realizar un placement tentativo.



# Pcbnew - Configurando el Stackup



## Stackup: Qué se define en KiCad?

- Cantidad y disposición de layers.
- Layers de cobre: Tipo de layer (Power, señal o mixto).
- Espesor final de la placa.

Pcbnew → Design Rules → Layers Setup

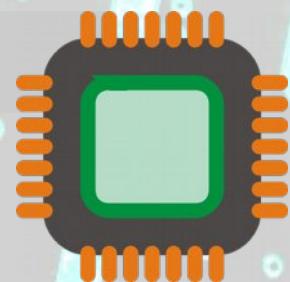
## Stackup: Qué configurar en KiCAD?

- Espesor de la placa (para vista 3D).
- Cantidad de layers de cobre.
- Nombre para cada layer.
- Tipo de layer: Signal, Power, Mixed.
- Componentes solo arriba o en ambas capas.
- Habilitar los layers que se utilizarán:
  - F.Paste: Pasta de estaño (TOP).
  - F.SilkS: Serigrafía (TOP).
  - F.Mask: Máscara antisoldante (TOP).
  - B.Mask: Máscara antisoldante (BOT).
  - B.SilkS: Serigrafía (BOT).
  - B.Paste: Pasta de estaño (BOT).
  - Edge Cuts: Borde de la placa
  - Dwgs. User: Layer para documentación.

# Ejercicio 2

## Editor de Footprints

# Pcbnew - Bibliotecas de footprints



Pcbnew posee soporte para los siguientes módulos:

- KiCad legacy (Los viejos .mod)
- KiCad Pretty (Los nuevos .kicad\_mod).
- Eagle (sin verificar).
- gEDA footprint libraries (sin verificar).

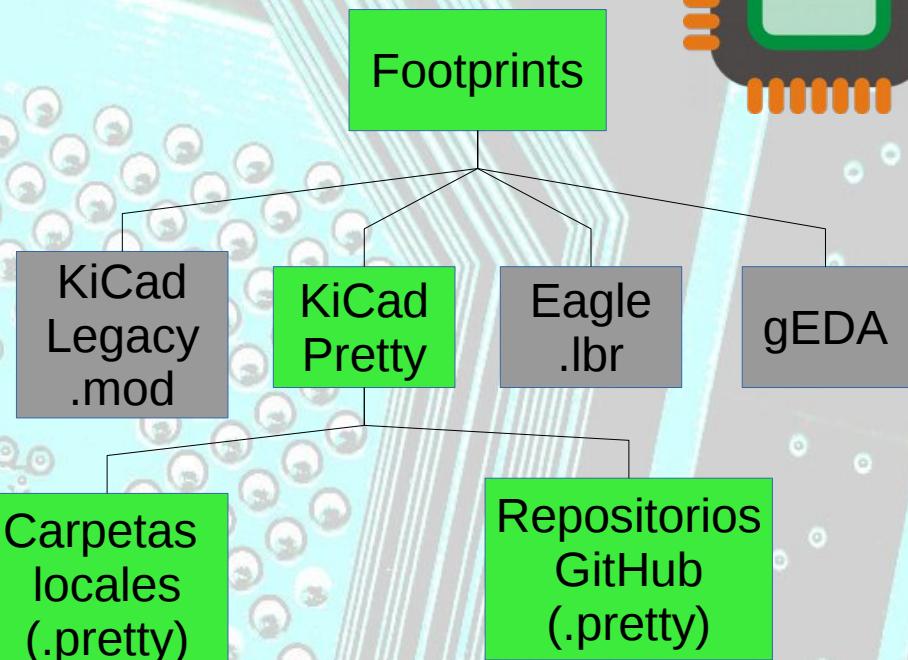
El Wizard permite fácilmente agregar:

- Directorios .pretty locales.
- Repositorios en Github.

Uno puede implementar sus propias bibliotecas en Github:

El Wizard permite bajar los repositorios seleccionados en forma local.

1) Ingresamos algunas (solo un par) bibliotecas de GitHub usando el Wizard.



<https://github.com/KiCad>

Search · kicad.pretty

GitHub Explore Features Enterprise Pricing

Search

We've found 151 repository results

Repositories 151

Code 2,836

Issues 183

Users

KiCad/kicad-library

The schematic and 3D libraries supported by KiCad team. Note that the footprint libraries are the \*.pretty repos themselves.

Updated 13 hours ago

Languages

Shell 20

Python 3

KiCad 1

Eagle 1

stormbard/Keyboard.pretty

KiCad Module Repository for keyboard PCB creation.

# Ejercicio 2 - Cambio de footprint

## MALAS NOTICIAS:

Desde el proyecto CIAA nos piden que utilicemos footprints SMD 1206 que se puedan soldar fácilmente a mano en vez del 0603 que hemos usado!



Para incursionar en el editor de footprints realizaremos el encapsulado SMD 1206 dese cero, para luego reemplazarlo en nuestro PCB.

1206!

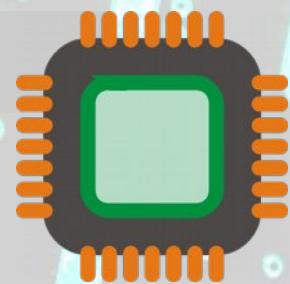
1206



Foto: Raúl Ferrari / Télam

Imagen cortesía de <http://www.telam.com.ar/>

# Pautas de footprints del proyecto KiCad



Como para los símbolos, el proyecto KiCad plantea pautas para los footprints:  
<https://github.com/KiCad/kicad-library/wiki/Kicad-Library-Convention>

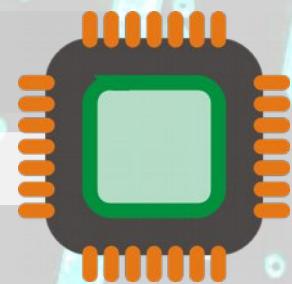
## General Rules for Footprints

- **Follows datasheet recommendation** unless intentional variation, for example longer pads for hand soldering.
- **Pad 1 is on the left first**, then at the top, except at the top for PLCC (IPC-7351).
- For through-hole components, footprint anchor is set on pad 1.
- For surface-mount devices, **footprint anchor is placed in the middle** with respect to device lead ends (IPC-7351).
- **Silkscreen is not superposed to pads, its outline is completely visible after board assembly**, uses 0.15mm line width and provides a reference mark for pin 1 (IPC-7351).
- Courtyard line has a width 0.05mm. This line is placed so that its clearance is measured from its center to the edges of pads and body, and its position is rounded on a grid of 0.05mm.
- Courtyard clearance is 0.25mm except for components smaller than 0603 at 0.15mm, connectors, SMD canned capacitors and crystals at 0.5mm and BGA at 1.0mm. (IPC-7251, IPC-7351B)
- **Cannot be duplicated to match a different pin ordering.**  
This is to be handled in the symbol libraries.
- **Value and reference have a height of 1mm.**

## Footprint properties

- Footprint name must match its filename. (.kicad\_mod files)
- Doc property contains a full description of footprint.
- Keywords are separated by spaces.
- **Value is filled with footprint name and is placed on the fabrication layer.**
- Attributes is set to the appropriate value, see tooltip for more information.
- All other properties are left to default values. (Move and Place: Free; Auto Place: 0 and 0, Local Clearance Values: 0)
- **3D Shape ".wrl" files are named the same as their footprint and are placed in a folder named the same as the footprint library replacing the ".pretty" with ".3dshapes".**

# Pautas de footprints del proyecto KiCad



También se define una convención para los nombres de los footprints:

## Names for footprints of Surface-Mount Devices (SMD)

- Specific package feature first, not separated by anything. Ex: "TSSOP".
- Package name, numbers separated from letters using hyphen. Ex: "SOT-89".
- Variation of package, separated by another hyphen. Ex: SOT-23 with 5 pins: "SOT-23-5", Exposed pad under package: "QFP-48-1EP".
- If it's a manufacturer-specific package, name can be appended, separated by an underscore.
- Any modification to the original footprint, indicated by appending the reason. Ex: longer pads used to facilitate hand soldering of a QFN component: "QFN-52\_HandSoldering".

## Names for footprints of common devices, such as resistors, capacitors, etc

- Name of part, may be shortened for common components. ex: "Cap", "Socket\_Strip", etc.
- Dimension, which may include at its end the positioning. Ex: "5x7mm\_Horiz", "1x02\_Angled".
- Pad distance, in the form of an RM rating.
- Any modification to the original footprint, indicated by appending the reason.

## Names for footprints of specific devices

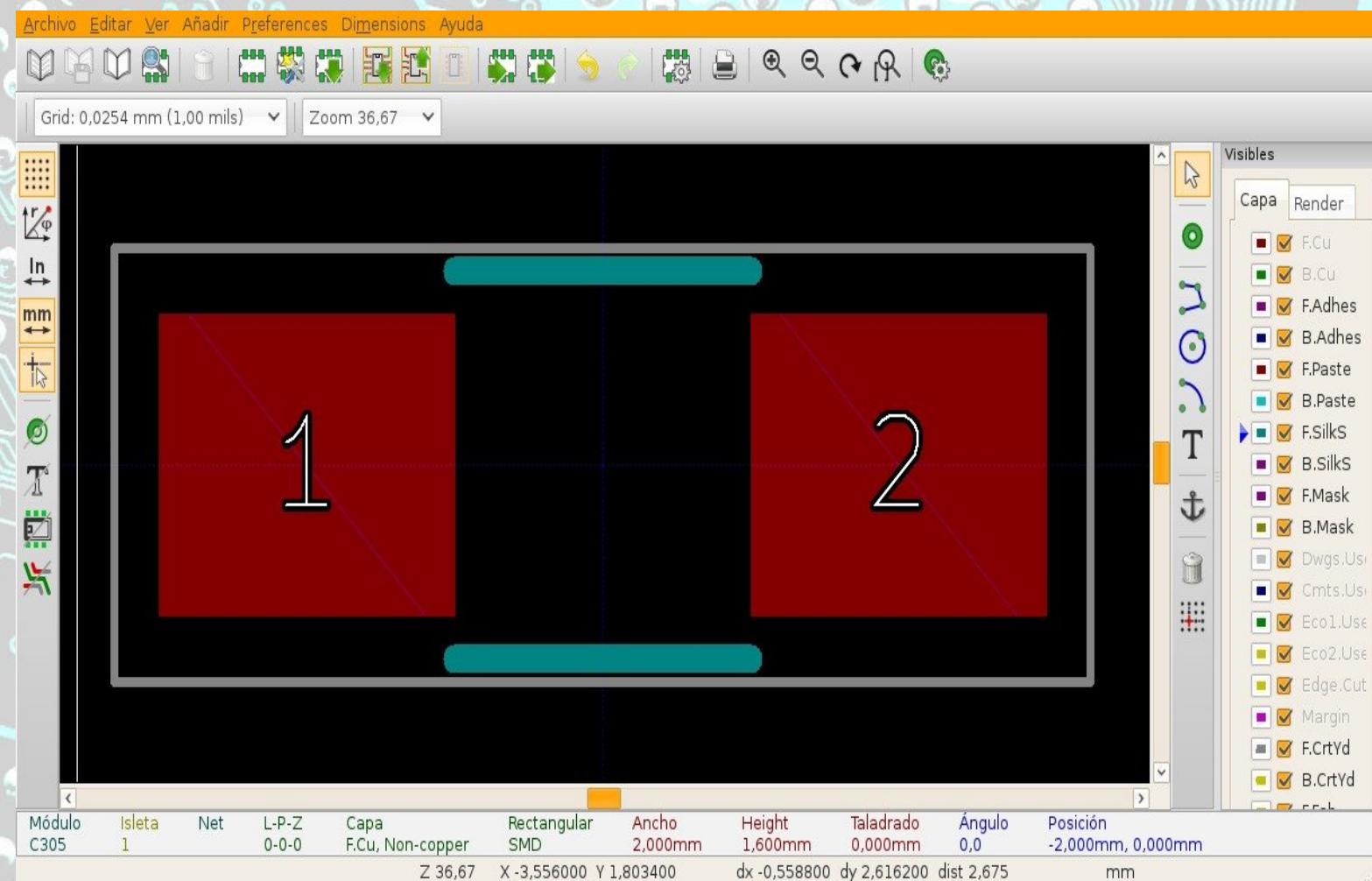
- Name of part.
- Part number. Ex: "Oscillator\_SI570"
- Any modification to the original footprint, indicated by appending the reason.

*Consultar la última versión de las pautas para símbolos y footprints:  
<https://github.com/KiCad/kicad-library/wiki/Kicad-Library-Convention>*

## Ejercicio 2 - Creación de footprints

Vamos a realizar un footprint SMD para soldar a mano un 1206. Los pads son un poco más grandes que el 1206 normal.

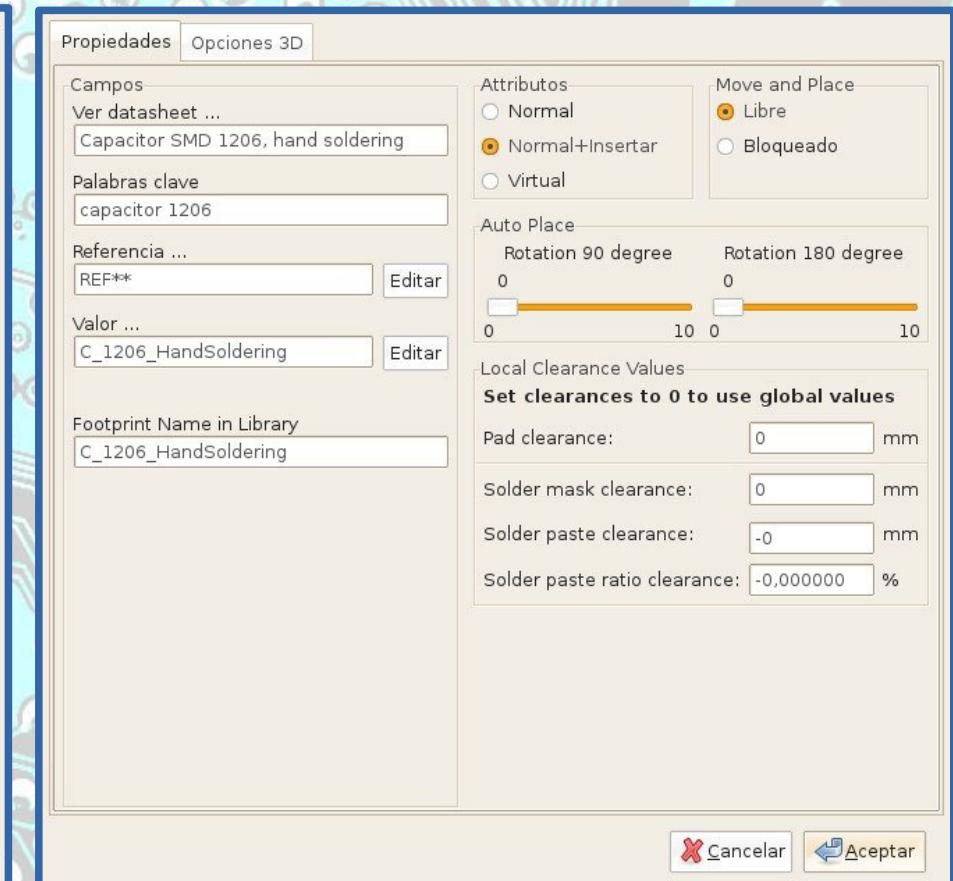
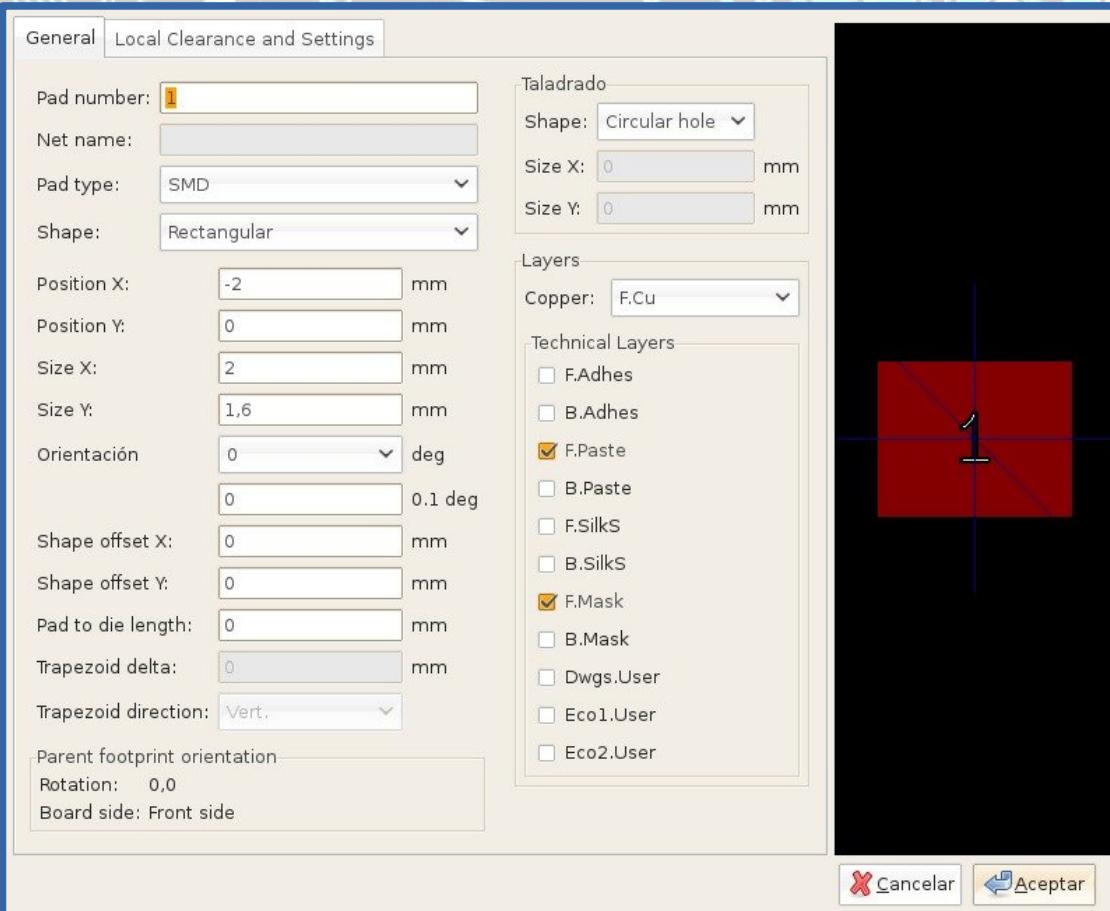
- 1)Abrir el editor de módulos.
- 2)Nuevo footprint.



# Ejercicio 2 - Creación de footprints

Crearemos un footprint para SMD 1206 con facilidades para soldar a mano.

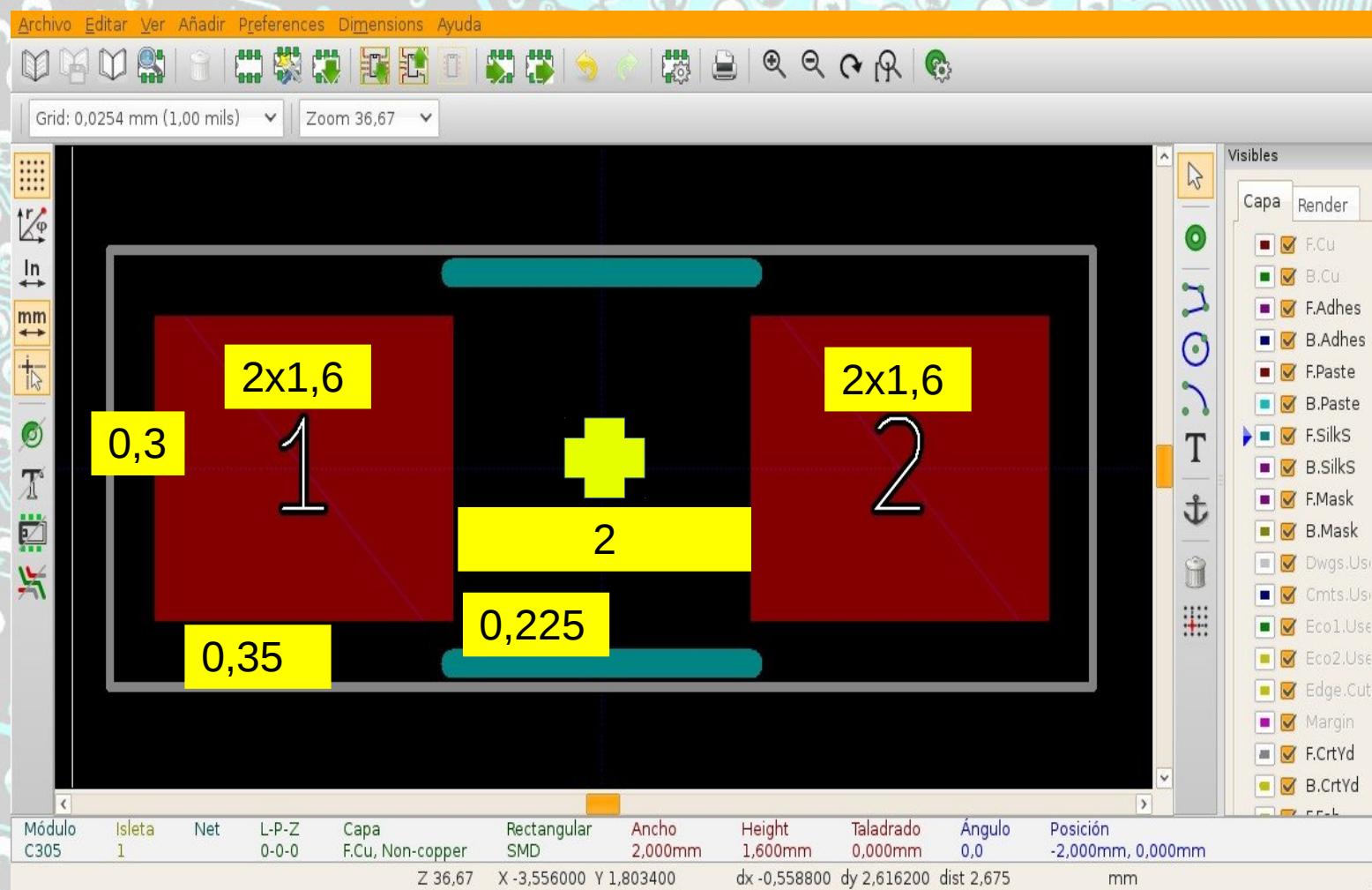
- 1) Crear los pads 1 y 2.
- 2) Completar las propiedades.



## Ejercicio 2 - Creación de footprints

Dimensiones en mm.

- 1) Ubicar los pads.
- 2) Dibujar y ubicar los demás elementos (ver información adicional en el siguiente slide).

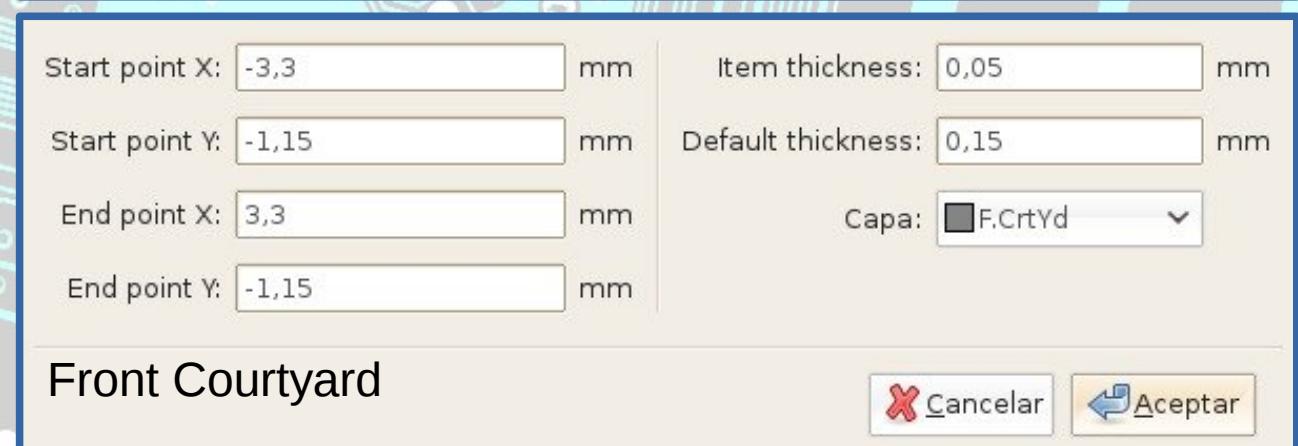
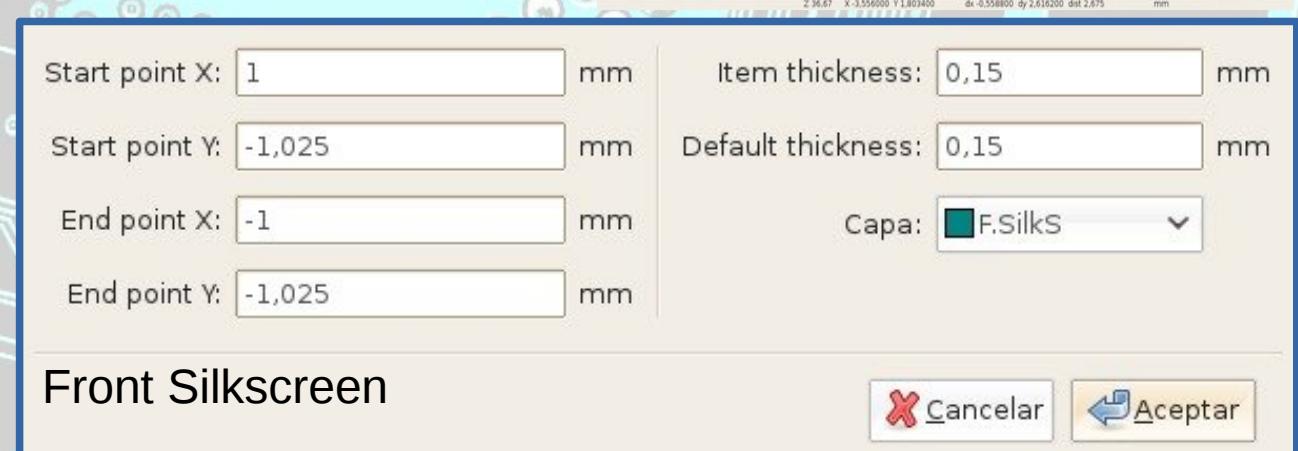
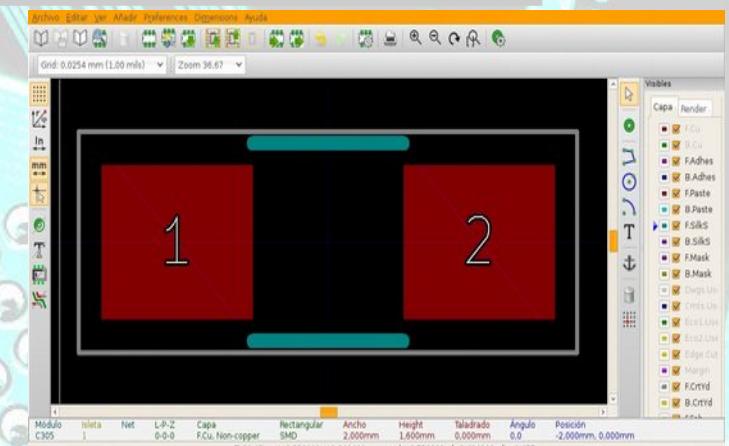


Origen

# Ejercicio 2 - Creación de footprints

Líneas de serigrafía y courtyard.

- 1)Dibujar el courtyard y la serigrafía.
- 2)Una vez finalizado exportar el footprint al directorio ej2.pretty.
- 3)En el esquemático o en el Cvpcb asociar los capacitores al footprint nuevo.
- 4)Exportar el netlist.



# Ejercicio 2

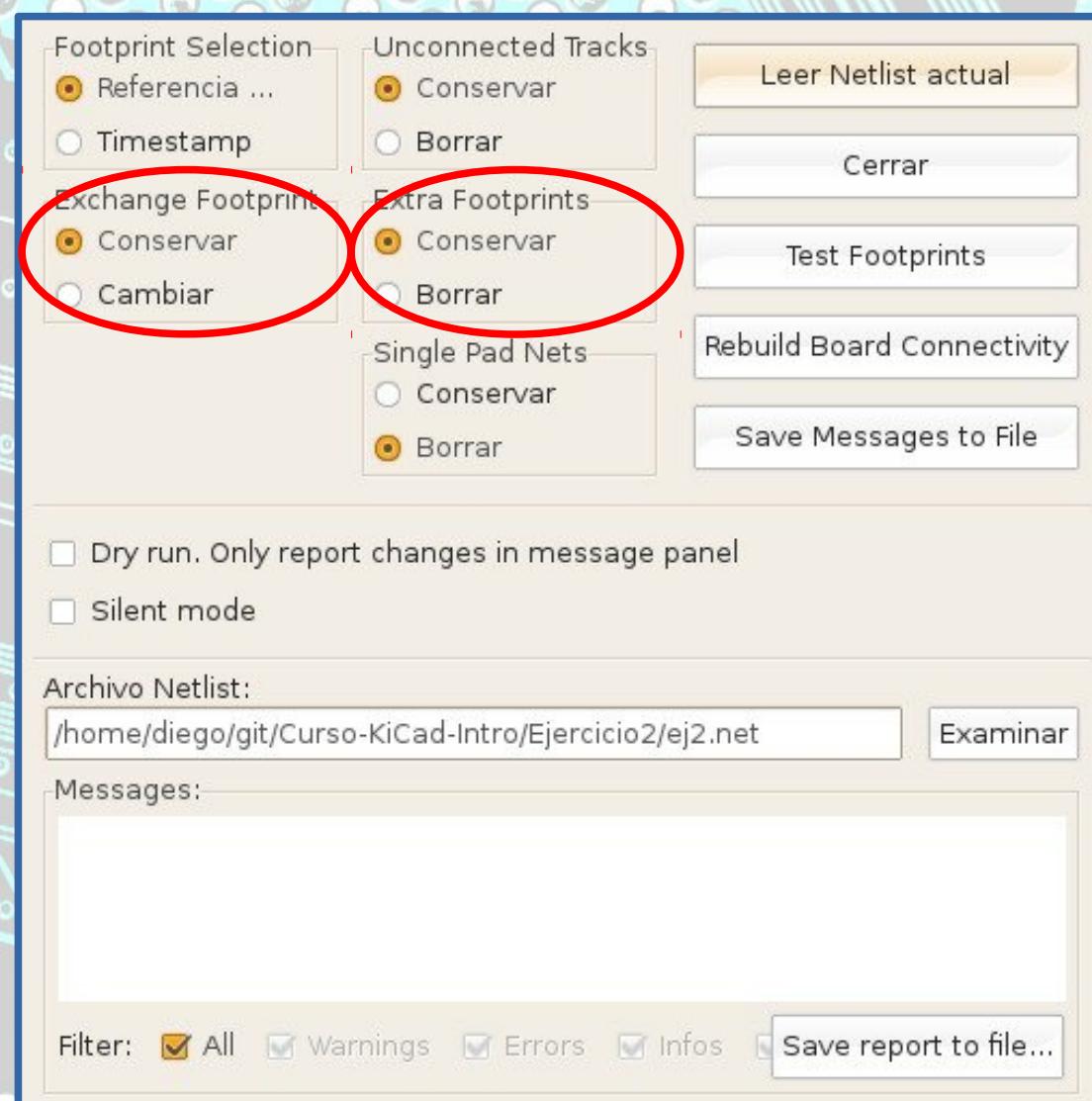
## Editor de PCB

## Ejercicio 2 - Reemplazo de footprints

En el Pcbnew ingresamos el netlist nuevo.

Cambiar las opciones por defecto de “Exchange Footprint” y “Extra Footprint”. Probaremos primero la opción “Dry run”.

- 1) Ir al diálogo de ingreso de netlist.
- 2) Habilitar “Dry run”.
- 3) Configurar para cambiar los footprints y borrar los footprints que sobren (esta última opción no es necesaria en este ejemplo).
- 4) Leer netlist y analizar la ventana de mensajes.
- 5) Deshabilitar “Dry run” y realizar la operación.
- 6) Realizar nuevamente la ubicación de componentes.

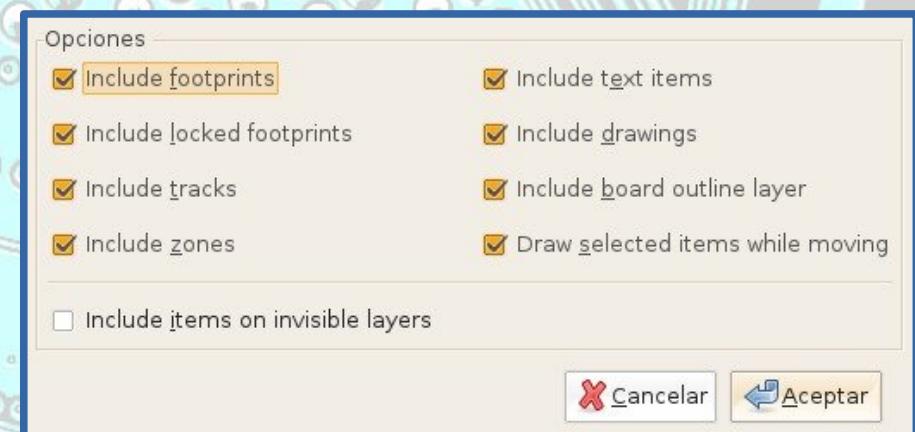


# Pcbnew - Operaciones con bloques

En el editor de PCB podemos realizar algunas operaciones de bloques como mover, copiar, espejar, girar y borrar.

Para realizar las operaciones de bloques:

- 1) Presionar el botón izquierdo de mouse en la primer esquina del bloque.
- 2) Manteniendo presionado, mover el mouse hasta la otra esquina del bloque a definir.
- 3) Soltar el botón izquierdo y aparecerá un diálogo para seleccionar qué elementos del bloque considerar y cuales no.



**Tip:** Para desrutear un circuito o una parte del mismo, incluir solo los tracks.

# Pcbnew - Operaciones con bloques

Luego del diálogo de selección de objetos.

## MOVER BLOQUE

- 1) Simplemente mover el mouse para mover el bloque.
- 2) Click izquierdo para colocar el bloque.

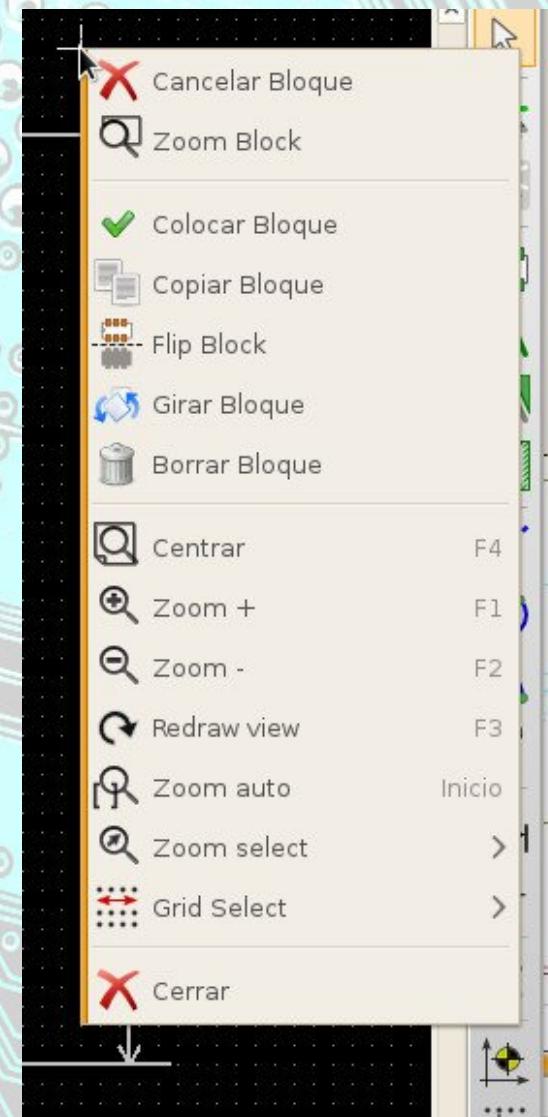
## COPIAR BLOQUE

- 1) Mover el bloque seleccionado a la posición donde se desea la copia (igual que mover bloque).
- 2) Botón derecho y seleccionar “Copiar bloque”.

## ESPEJO, GIRO y BORRAR

- 1) Botón derecho y elegir la opción correspondiente.

- 1) Probar operaciones de bloques, usando el “deshacer” luego de cada una.



## Ejercicio 2 - Ruteo

Elegiremos la tecnología de 12 mils para nuestro diseño.

- 1) Configurar ancho de traza, vías y márgenes según la tabla del fabricante.

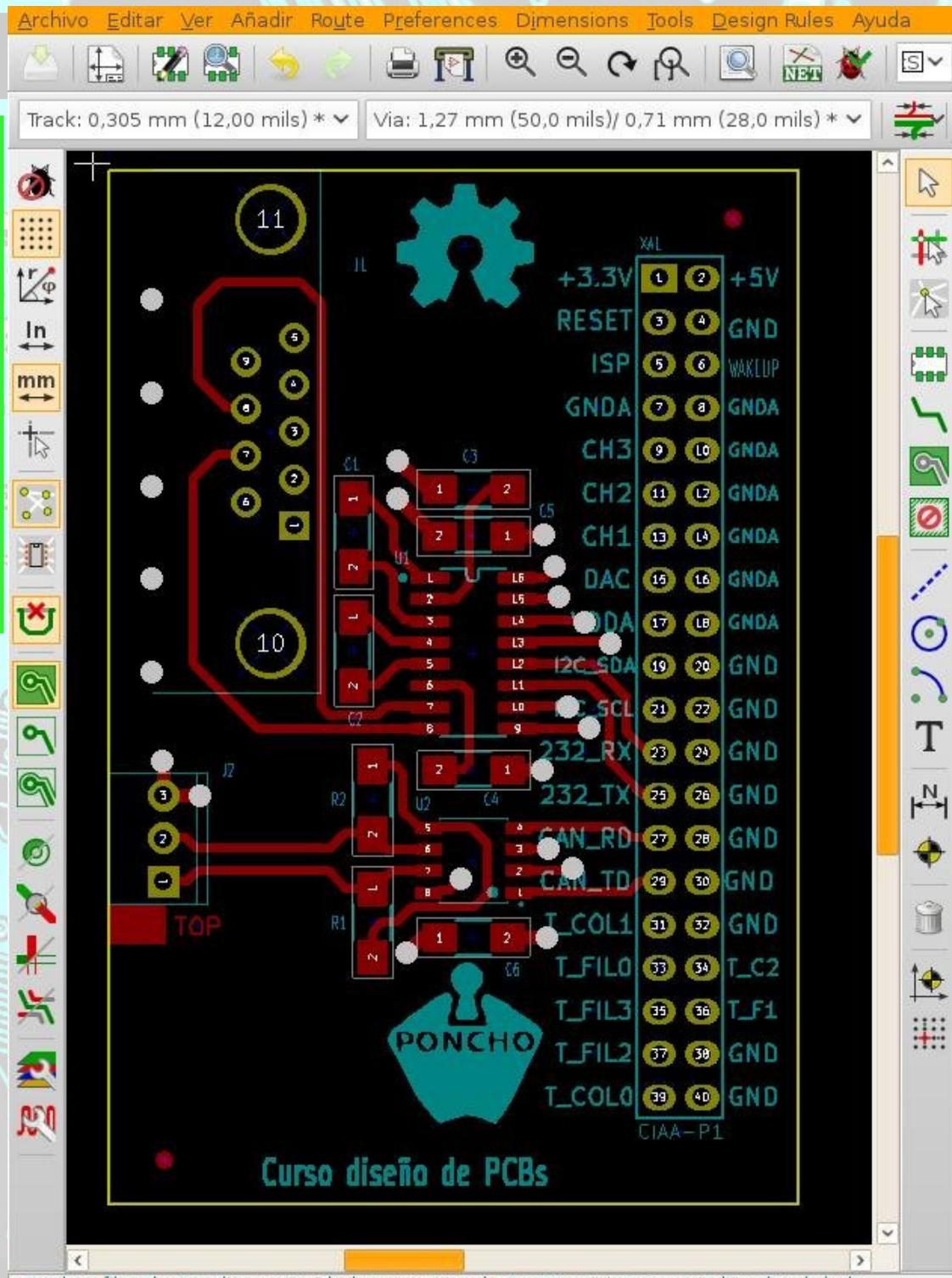
### Estándares de fabricación

TECNOLOGIA	12 MILS	10 MILS	8 MILS	6 MILS
DIAMETRO DE AGUJERO	28 (0.70 mm)	20 (0.50 mm)	16 (0.40 mm)	16 (0.40 mm)
DIAMETRO DE PAD O VIA	50 (1.27 mm)	40 (1.00 mm)	32 (0.80mm)	28 (0.70mm)
ANCHO DE TRAZA	12 (0.30mm)	10 (0.25 mm)	8 (0.20 mm)	6 (0.15mm)
SEPARACION ENTRE TRAZAS	12 (0.30mm)	10 (0.25 mm)	7 (0.18 mm)	6 (0.15mm)
SEPARACION ENTRE TRAZA Y PAD/VIA	10 (0.25 mm)	8 (0.20 mm)	6 (0.15mm)	6 (0.15mm)
DISTANCIA DE COBRE A BORDE	12 (0.30mm)	12 (0.30mm)	12 (0.30mm)	12 (0.30mm)
ALTURA - TRAZO DE LETRAS	48-8	36-6	30-5	30-5

Imagen cortesía de <http://www.mayerpcb.com/>

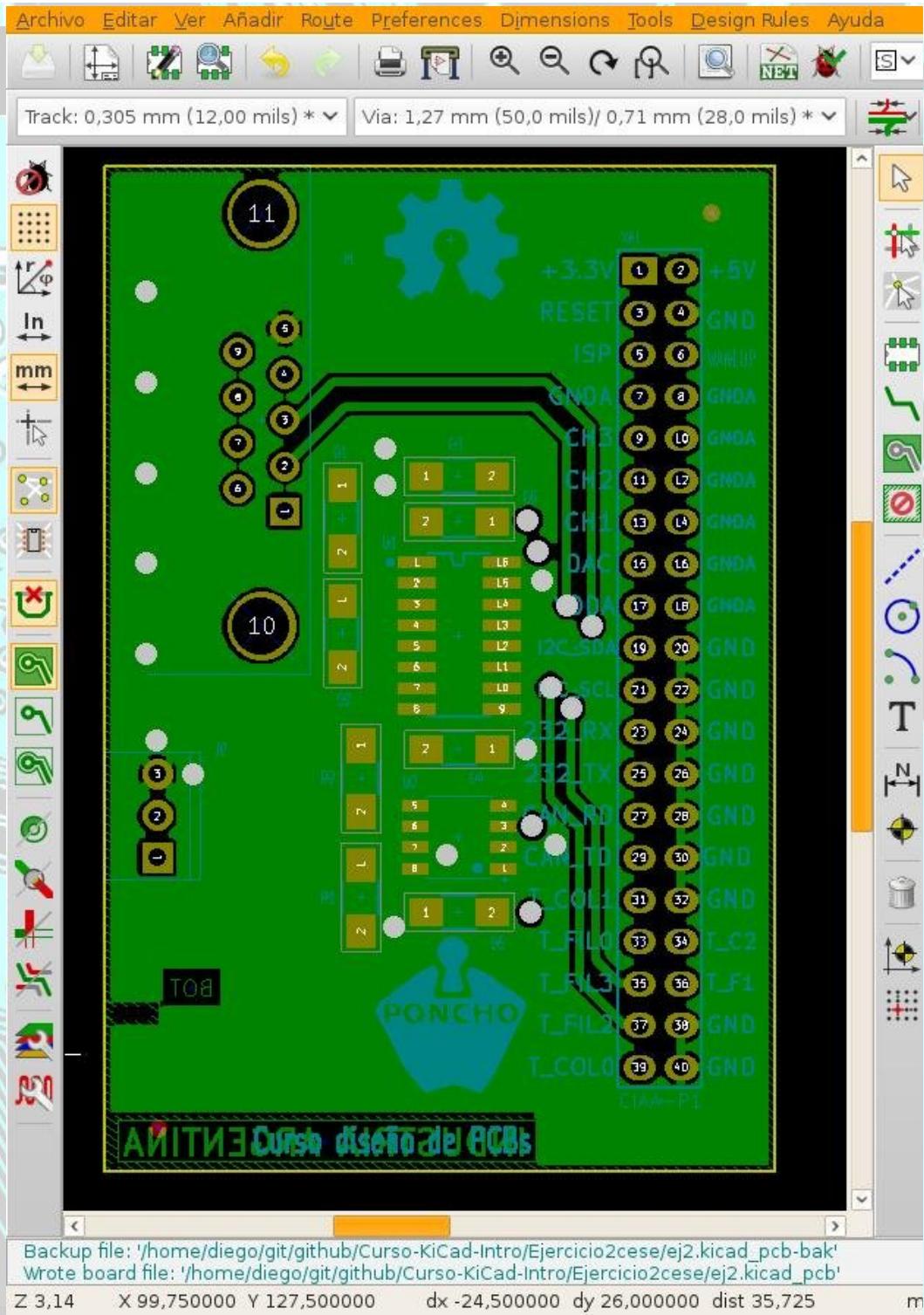
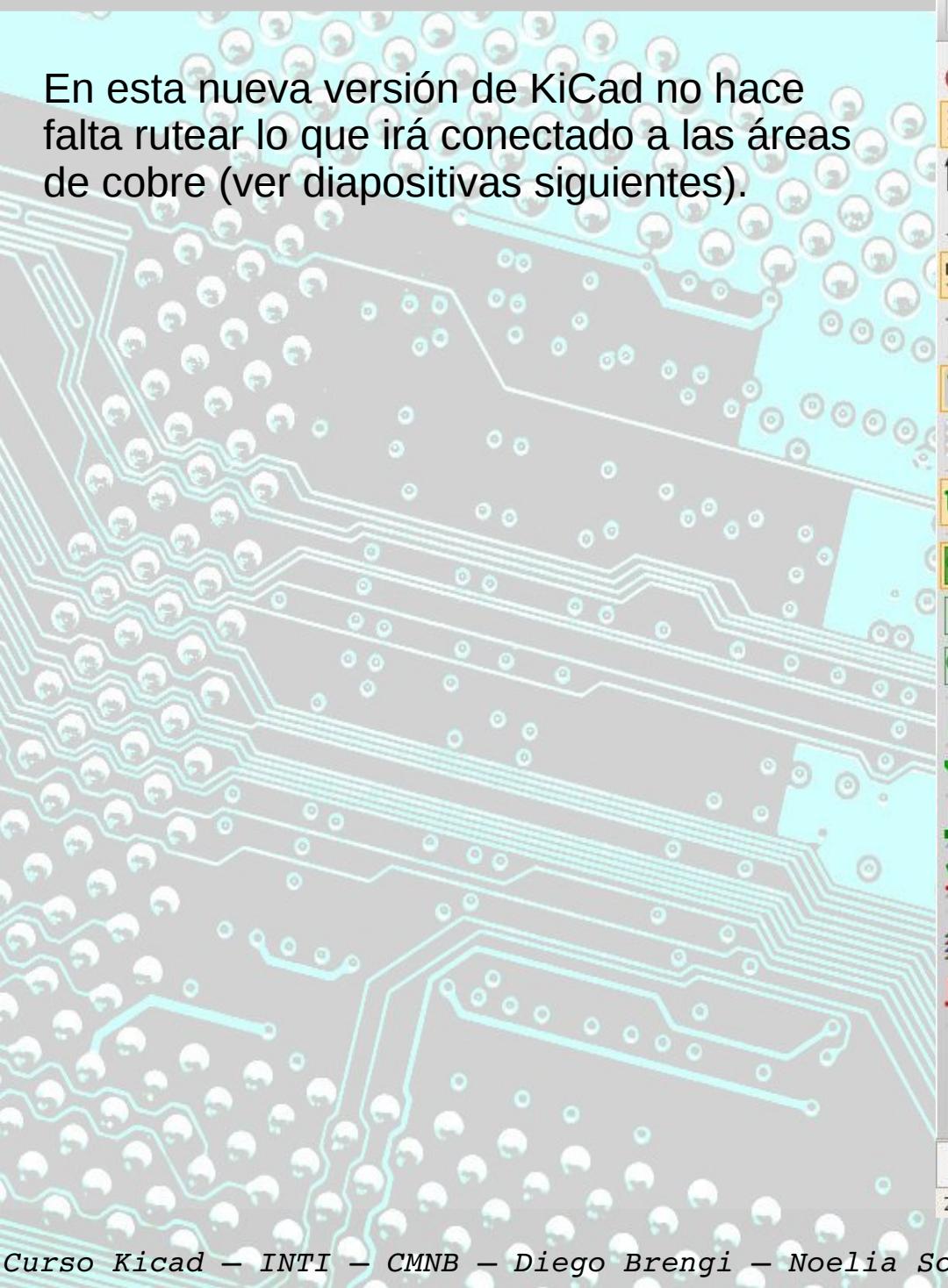
# Ejercicio 2 - Ruteo

- 1) Realizar el ruteo, usando líneas a 45 grados.
- 2) Investigar las opciones de mover nodo, mover segmento, mantener pendiente, etc.
- 3) Las líneas del CAN se rutean en forma tradicional.
- 4) Los planos de tierra se explican en las siguientes diapositivas. No conectar las tierras ni las alimentaciones.



## Ejercicio 2 - Ruteo

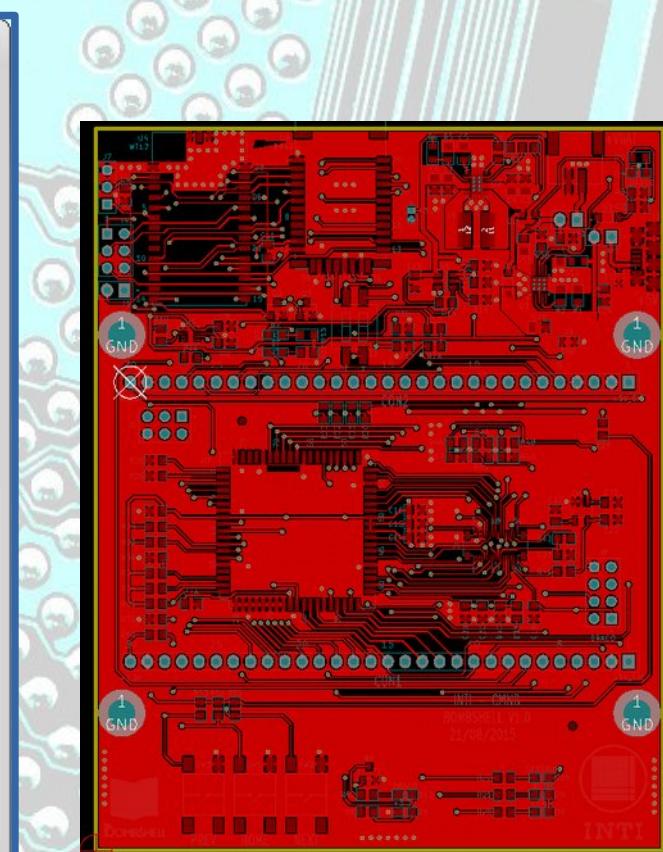
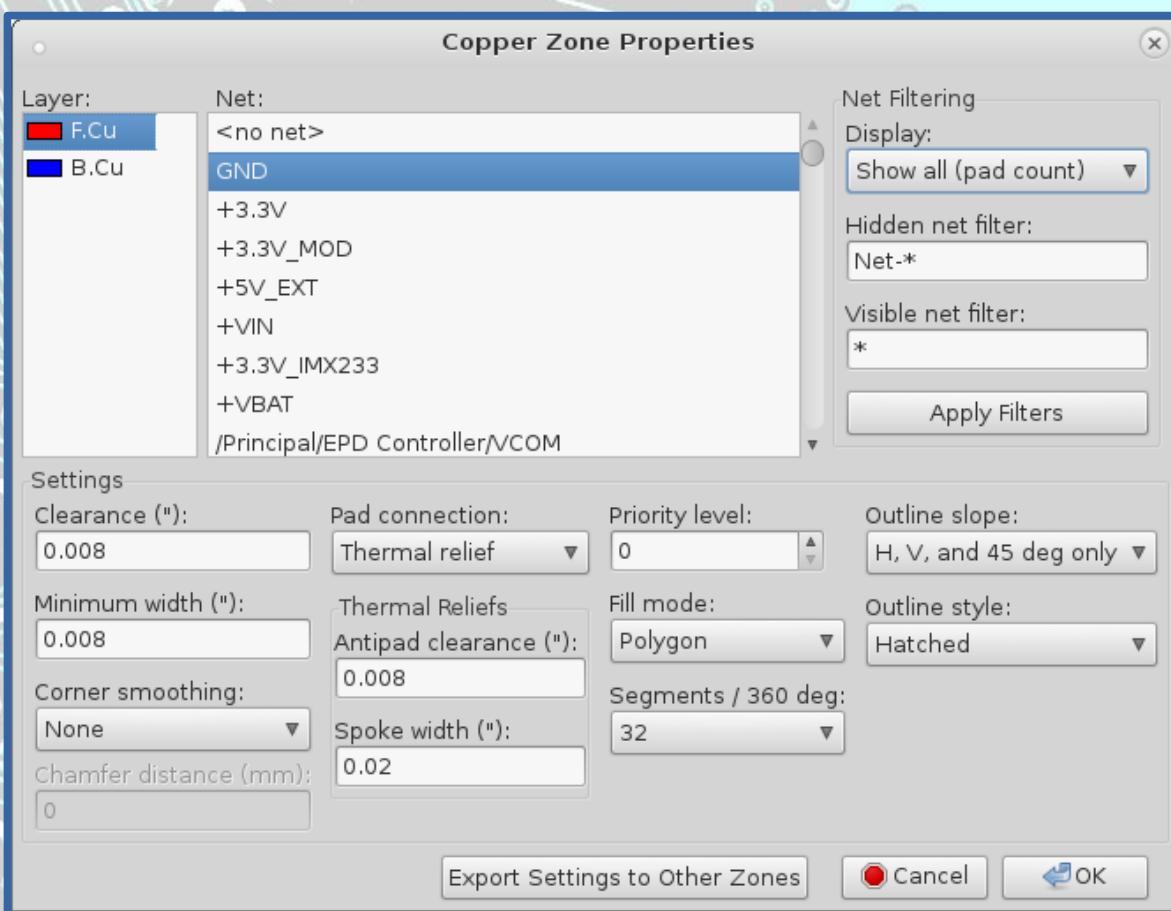
En esta nueva versión de KiCad no hace falta rutear lo que irá conectado a las áreas de cobre (ver diapositivas siguientes).



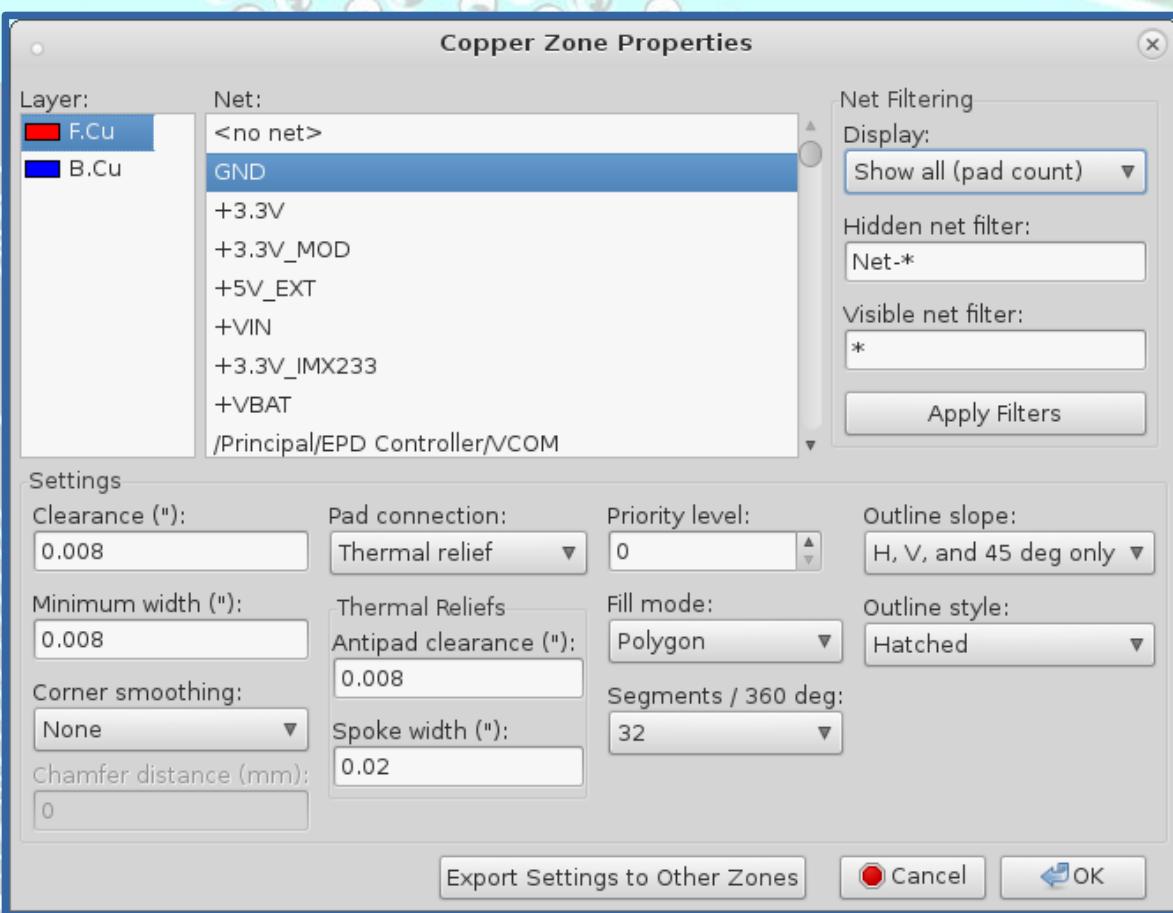
# Pcbnew - Áreas de cobre (Copper zones) I

## Creación de áreas de cobre

- 1) Activar cualquier layer de cobre y clickear en el botón **Agregar Zona de Cobre**.
- 2) Seleccionar Layer y Net.
- 3) Configurar todos los parámetros deseados: Clearance, Minimum Width, Pad Connection, Fill mode, Priority level.
- 4) Trazar los bordes de la zona y finalizar con doble click.
- 5) Para rellenar hacer click derecho en el borde del área e ir a **Zones → Fill Zone**.

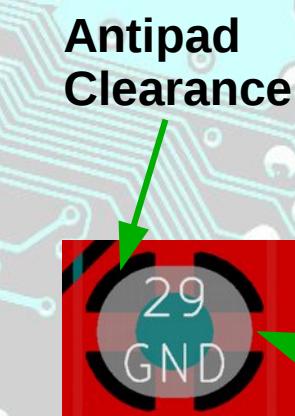


# Pcbnew - Áreas de cobre (Copper zones) II



**Clearance:** Poner un valor un poco superior al clearance general de la placa.

**Minimum Width:** Se utiliza para limitar la formación de áreas de cobre muy pequeñas. No se debe agrandar mucho porque afecta a los thermal reliefs.



**Priority Level:** Se utiliza para crear zonas dentro de otras zonas. Poner en 0 para grandes zonas y en 1 ó más para sub-zonas.

**Fill mode:** Dejar en Polygons.

**Segments:** Dejar en 32, ya que no afecta la performance.

**Outline Slope:** Dejar en H, V y 45º.

**Outline Style:** Dejar en Hatched.

**Pad Connection:** Por lo general se utiliza Thermal Relief para que no se dificulte la soldadura de los componentes. En los casos de planos que se agregan para disipación, puede dejarse en Solid.

**Antipad Clearance:** Un valor muy grande puede impedir que se forme un thermal relief en un pad muy pequeño.

**Spoke width:** Debe ser mayor que el valor Minimum Width.

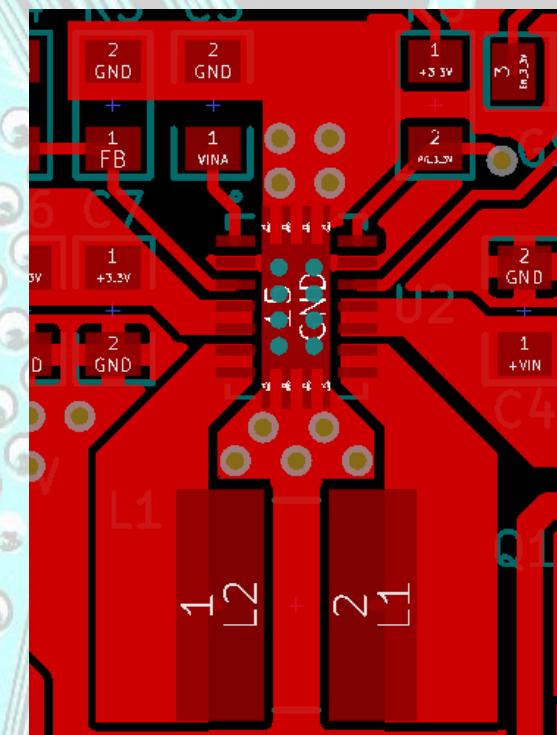
**Spoke Width  
(Track Width)**

# Pcbnew - Áreas de cobre (Copper zones) III

**Fill Zones:** Para reconstruir zonas usar la tecla '**B**'. Utilizar periódicamente para tener siempre actualizados los planos!

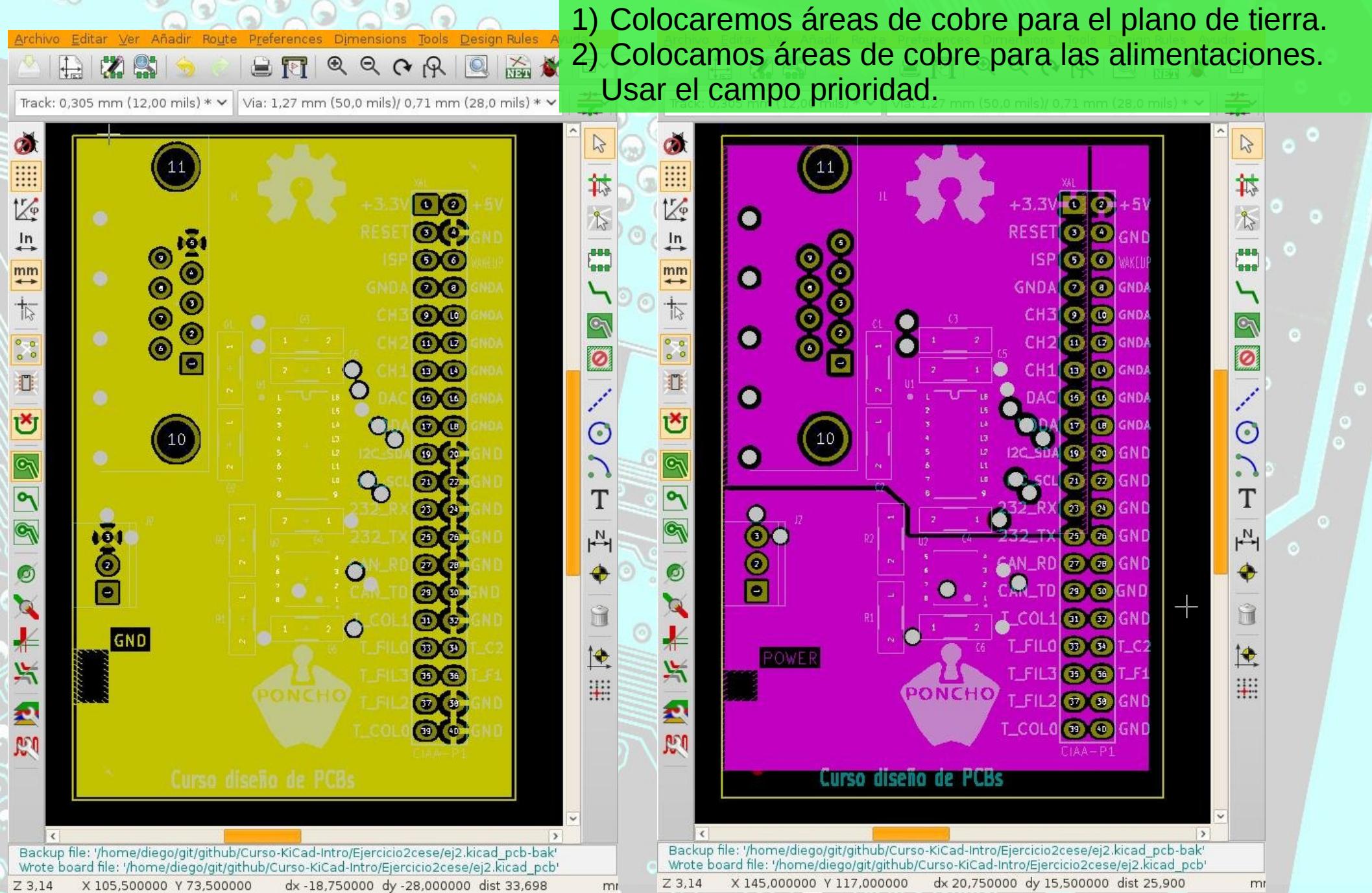
**Unfill Zones:** Para sacar temporalmente el relleno de las zonas, usar '**Ctrl+B**'. Es útil si se debe rutear nuevamente un sector del PCB o mover algún componente.

Para editar la forma de una zona existente se pueden crear y eliminar **corners (puntos de inflexión)**, esto resulta útil para perfeccionar el contorno de la zona!

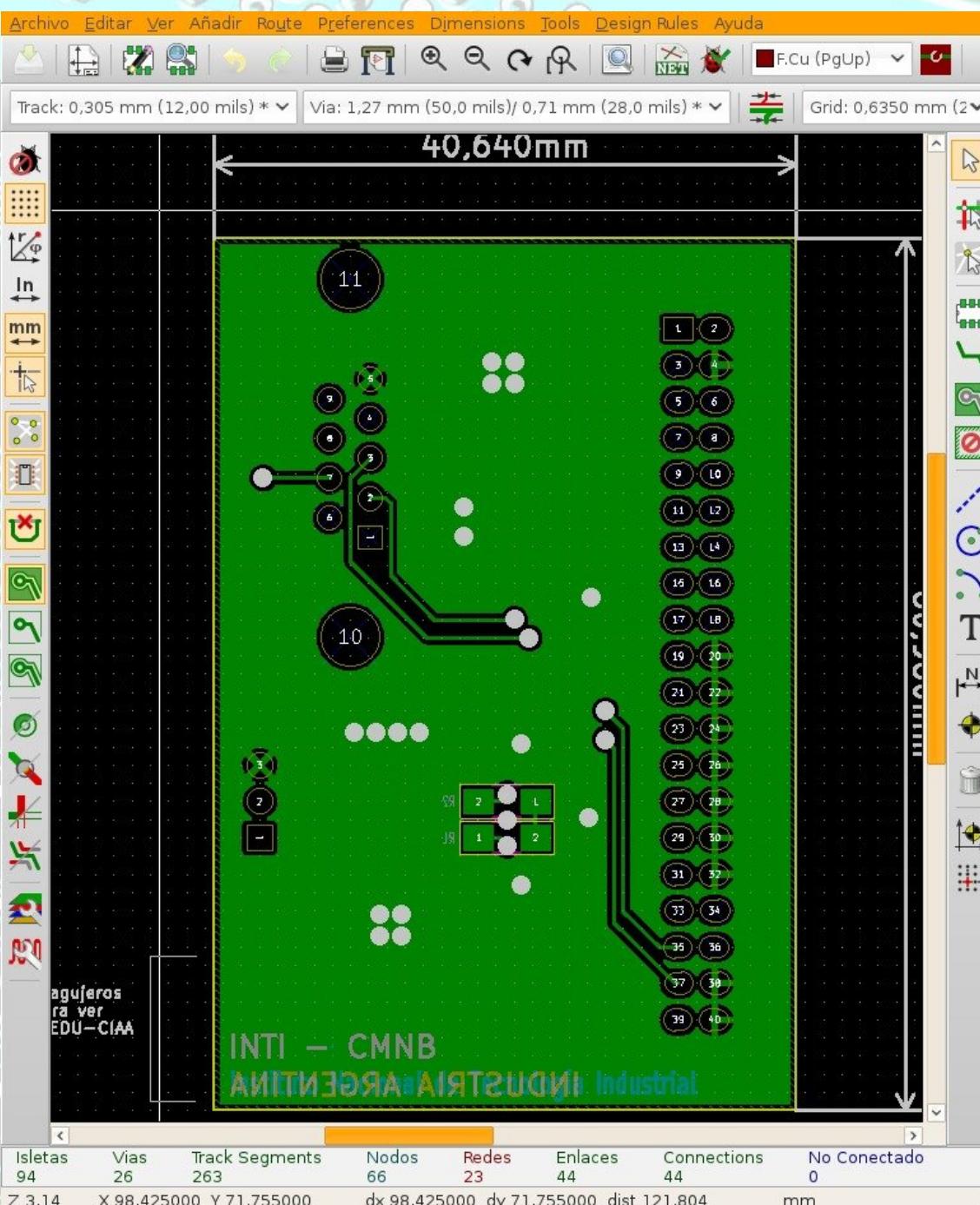


# Ejercicio 2 - Áreas de cobre y planos de tierra

- 1) Colocaremos áreas de cobre para el plano de tierra.
- 2) Colocamos áreas de cobre para las alimentaciones. Usar el campo prioridad.



# Ejercicio 2 - Áreas de cobre



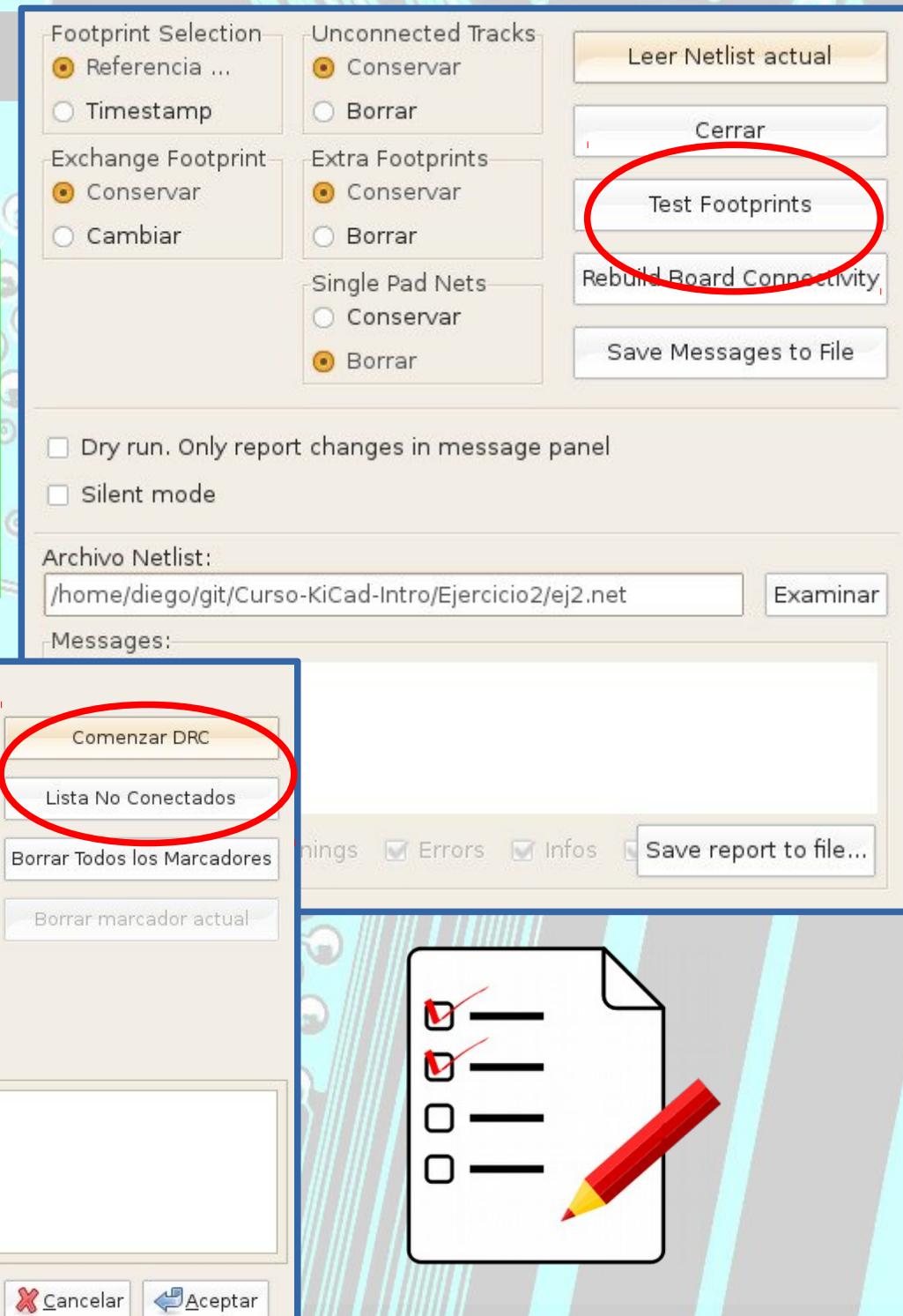
- 1) Colocar textos en serigrafía superior.
- 2) Colocar el texto INDUSTRIA ARGENTINA en la capa bottom de cobre.
- 3) Colocar logos.
- 4) No olvidar también acomodar la serigrafía (campo referencia).

TIP: Algunos textos adicionales a colocar:  
INDUSTRIA  
ARGENTINA, Empresa o institución, Fecha, SHA-1 corto o tag, URL, Autor/es, Proyecto, etc.

# Ejercicio 2 - Verificaciones

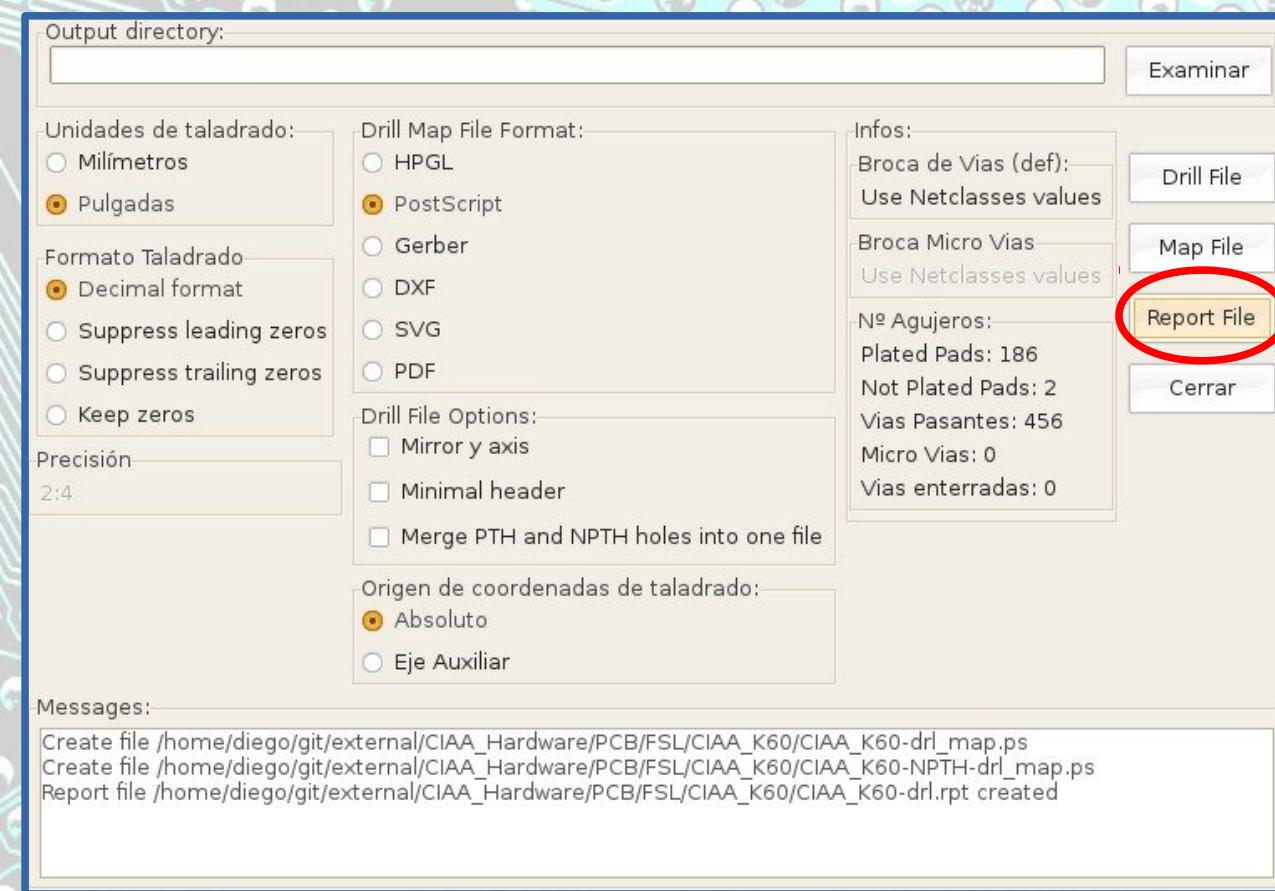
Una vez terminado el ruteo realizaremos algunas verificaciones simples.

- 1) En el diálogo de netlist usar el botón de "Test Footprints".
- 2) Correr el DRC y la opción de "Lista No conectados."
- 3) También se puede generar el netlist en el esquemático y leerlo nuevamente para asegurarnos que todo coincide.



# Taladrado: Reporte y mapa

Además de los archivos gerbers, se debe revisar el reporte de taladrado para saber los distintos agujeros que tenemos. Podríamos optimizar si es necesario.



Drill report for CIAA\_K60.kicad\_pcb  
Created on mar 01 dic 2015 22:25:47 ART

Drill report for plated through holes :

T1 0,41mm 0,016" (456 holes)  
T2 0,75mm 0,030" (4 holes)  
T3 0,90mm 0,035" (8 holes)  
T4 1,00mm 0,039" (20 holes)  
T5 1,02mm 0,040" (88 holes)  
T6 1,20mm 0,047" (8 holes)  
T7 1,40mm 0,055" (40 holes)  
T8 1,50mm 0,059" (10 holes)  
T9 1,60mm 0,063" (2 holes)  
T10 3,40mm 0,134" (2 holes)  
T11 4,00mm 0,157" (4 holes)

Total plated holes count 642

Drill report for buried and blind vias :

Holes from layer TOP to layer GND :

Total plated holes count 0

Holes from layer GND to layer PWR :

Total plated holes count 0

Holes from layer PWR to layer BOT :

Total plated holes count 0

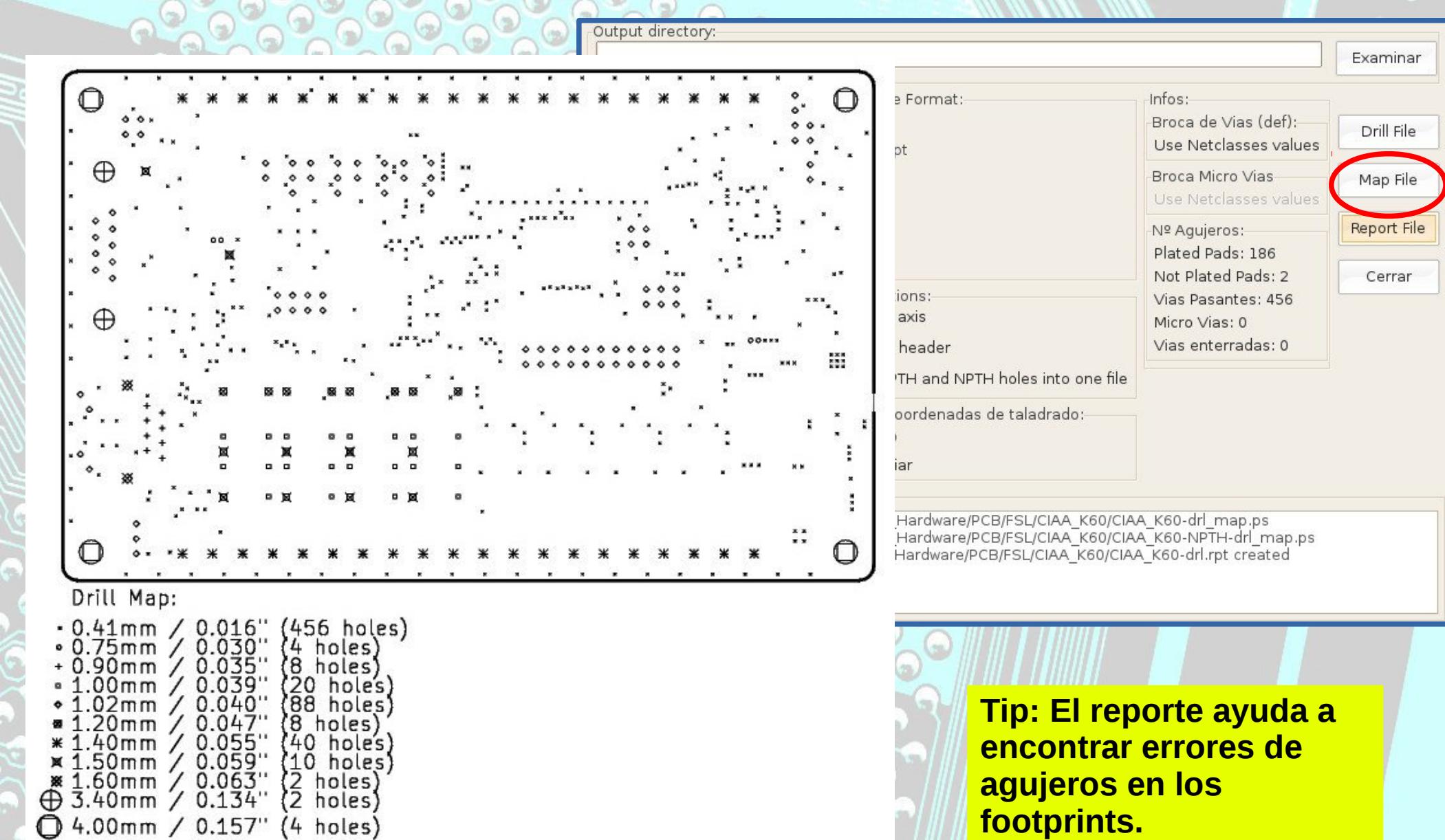
Drill report for unplated through holes :

T1 3,25mm 0,128" (2 holes)

Total unplated holes count 2

# Taladrado: Reporte y mapa

También podemos generar el reporte en forma de mapa.

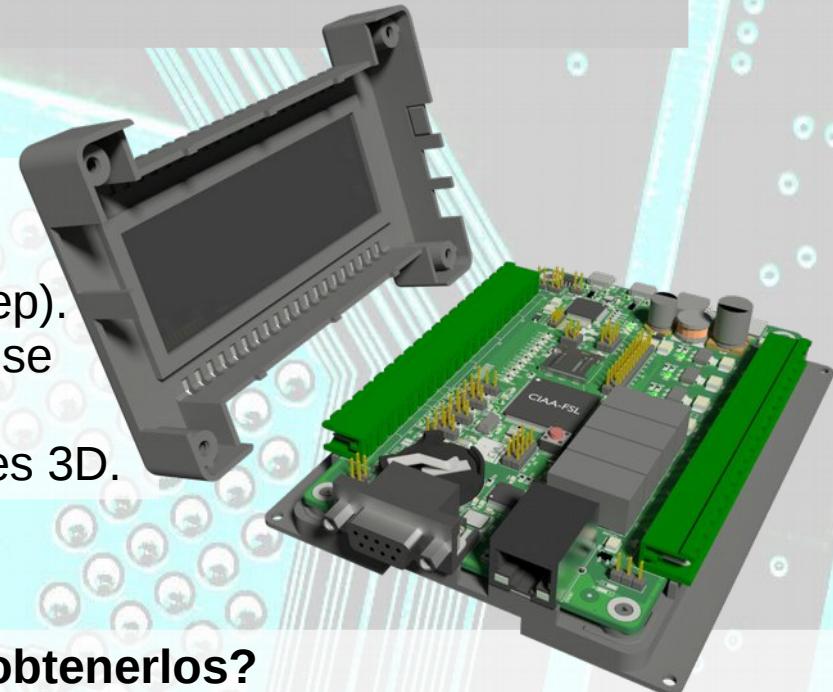
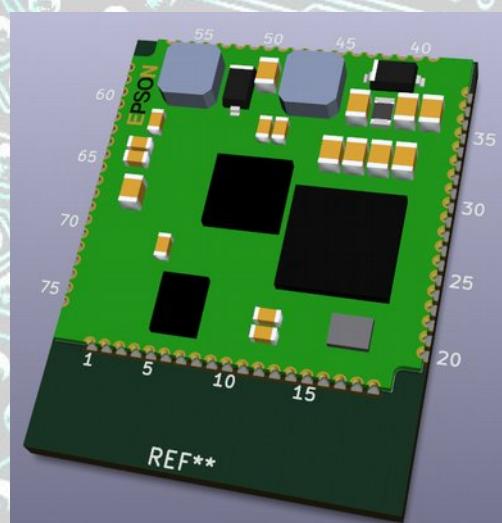


**Tip: El reporte ayuda a encontrar errores de agujeros en los footprints.**

# Pcbnew - Modelos 3D I

## Modelos 3D: Cuándo usarlos?

- Cuando se va a diseñar un gabinete o se necesita chequear uno existente (se debe tener el modelo step).
- Cuando se quiere verificar que los componentes no se toquen entre sí o no tapen la serigrafía.
- Para promocionar un producto o realizar animaciones 3D.



## Modelos 3D: De dónde obtenerlos?

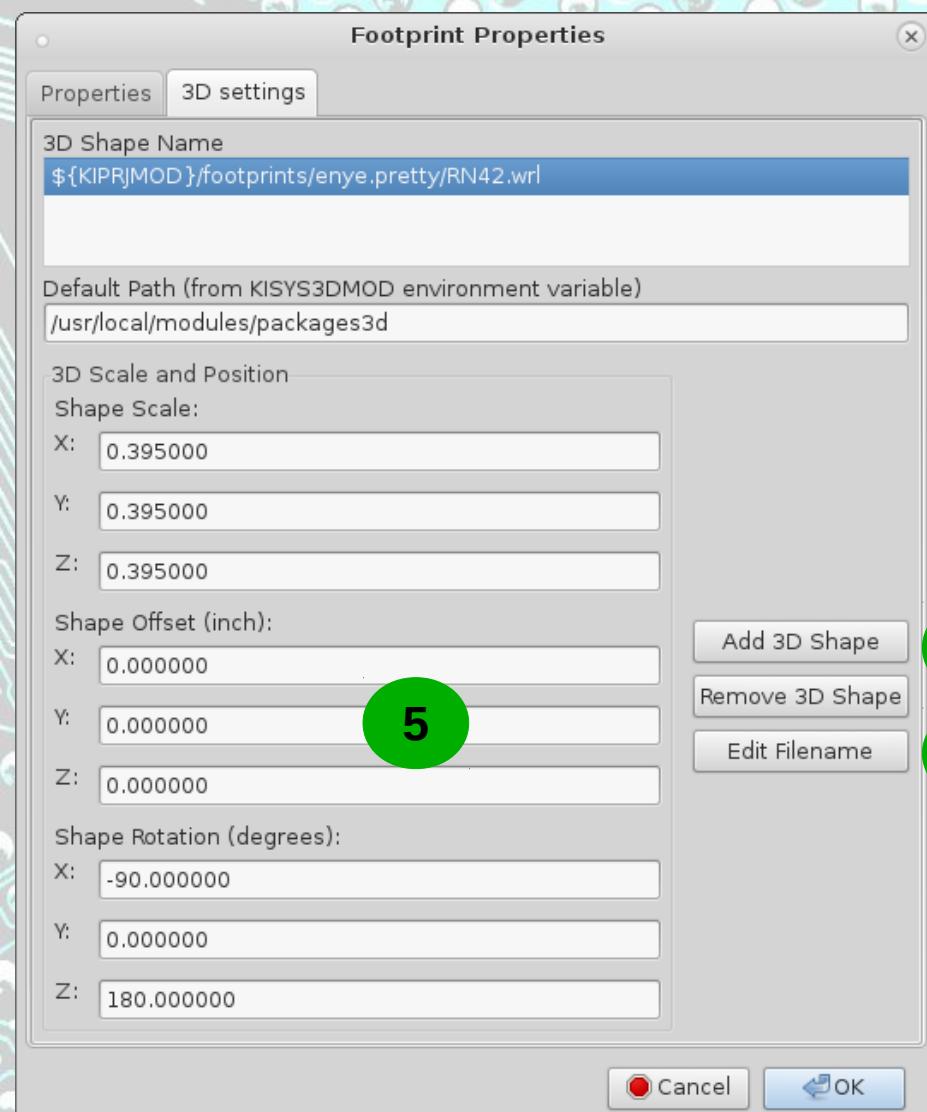
- Librerías estándar de KiCad  
<https://github.com/KiCad/kicad-library/tree/master/modules/packages3d>  
también en (/usr/local/share/kicad/modules/packages3d).
- Bajar el modelo IGES o STEP que provea el fabricante y convertirlo al formato VRML (.WRL) con FreeCad.
- Bajar el modelo de la página [www.3dcontentcentral.com](http://www.3dcontentcentral.com) (ver licencias).
- Crear el modelo 3D con FreeCad.



**Tip:** Usar FreeCad para convertir archivos o superponer modelos 3D.

# Pcbnew - Modelos 3D II

## Footprint Properties → 3D Settings

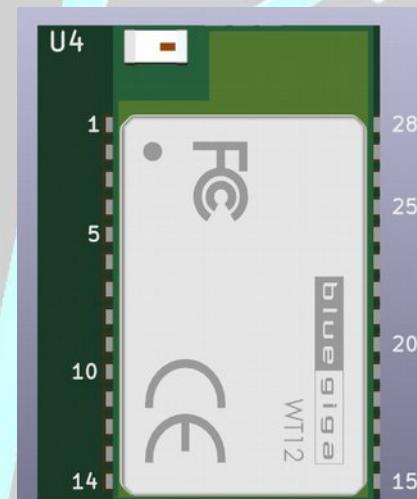


## Modelos 3D: Cómo asignarlos a un componente?

- 1) Copiar el archivo WRL en el directorio de 3D , por ej. /ej2.3dshapes/.
- 2) Seleccionar un componente, presionar la tecla 'E' y abrir el cuadro **3D Settings**.
- 3) Presionar **Add 3D Shape** y buscar el modelo.
- 4) Presionar **Edit Filename** y agregar en la ruta del archivo:  
**`\${KIPRJMOD}/ej2.3dshapes/`**
- 5) Si fuera necesario, escalar, desplazar o rotar el modelo.

**El cambio de ruta permitirá que al abrir el PCB desde otra ubicación, los modelos 3D se carguen correctamente!**

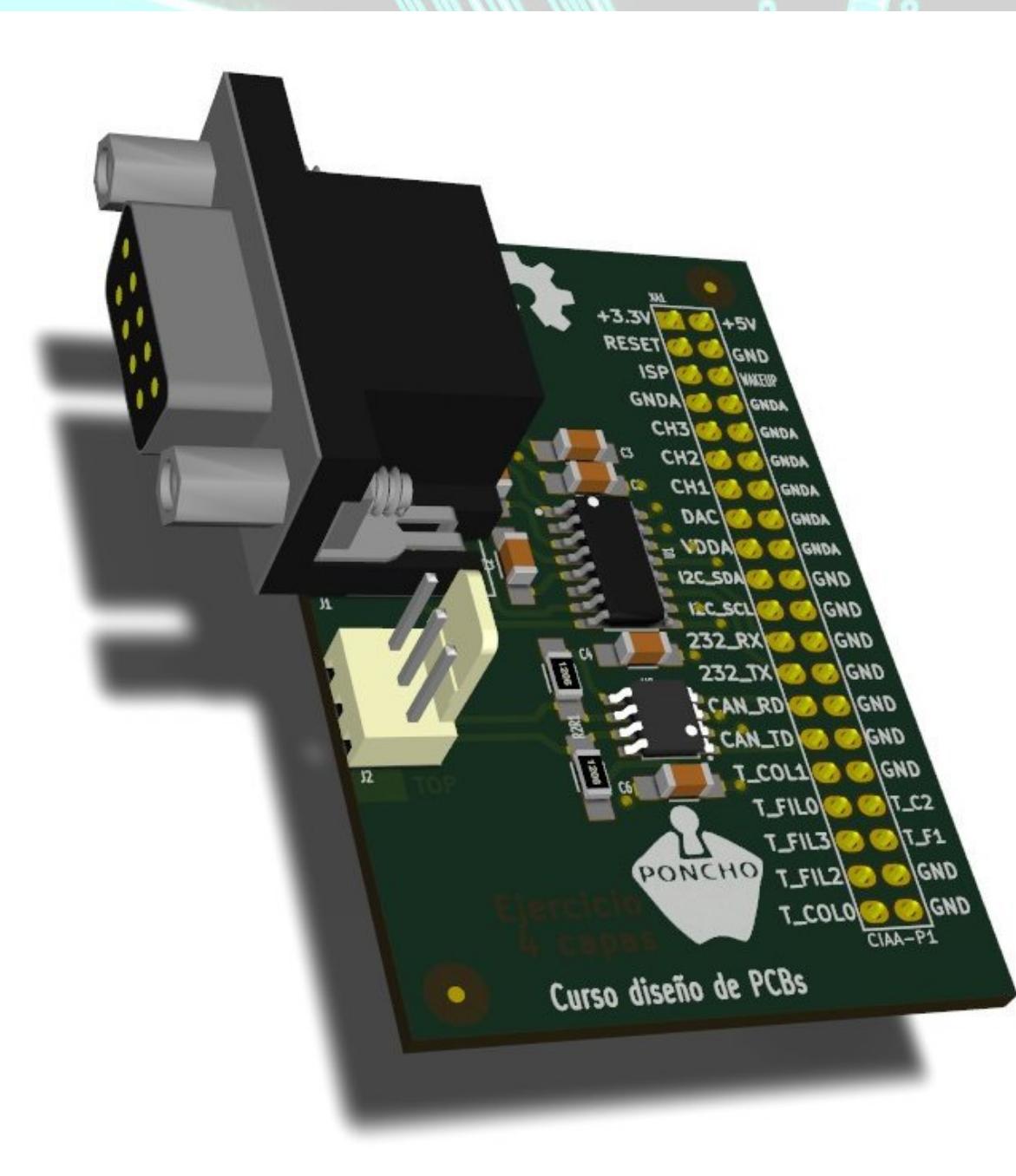
**Tip: Para la escala probar con el factor 0.395!**



# Ejercicio 2 - Modelos 3D

Colocar los modelos 3D de nuestro diseño.

- 1)Copiar los modelos 3D provistos en el directorio ej2.3dshapes.
- 2)Vincular los componentes según las instrucciones previas.
- 3)Ver la vista 3D.



# Ejercicio 2

## Ingreso a GIT

# Configuración de GIT

Antes de usar git conviene realizar algunas configuraciones globales básicas. Estas configuraciones aplicarán a todos los repositorios que utilicemos y se guardan en nuestro “home”.



```
$ git config --global user.name "Tu Nombre Completo"  
$ git config --global user.email Tu-email@servidor.com  
$ git config --global http.proxy http://IP_PROXY:PORT  
$ git config --global credential.helper "cache --timeout=28800"  
$ git config --global core.editor emacs
```

- 1) Realizar las configuraciones globales necesarias.
- 2) Verificar listando la configuración.

```
$ git config --list
```

# Ejercicio 2 - Ingreso a GIT - .gitignore

Ingresamos el proyecto nuevo a GIT.

El archivo .gitignore permite definir archivos que no queremos subir al repositorio y no queremos que GIT los reporte.

## ARCHIVO .gitignore

```
*.bak  
*.kicad_pcb-bak  
_autosave-*.*.kicad_pcb
```

1) Crear un archivo .gitignore.

## Comandos GIT

```
git add [Filename]  
git commit -m "msj"  
git status -s
```



# GIT - El merge y los conflictos con KiCad



¿COMO  
FUNCIONA el MERGE  
PARA ARCHIVOS  
KICAD?

El mecanismo de merge en git es muy útil para desarrollo de código y textos en general.

A pesar que los archivos KiCad son textos ASCII, resolver un conflicto en un archivo KiCad puede ser muy complicado y demanda tiempo, esfuerzo y conocimientos de su sintaxis. Especialmente cuando se realizan cambios en los gráficos o las posiciones de los objetos (ver adentro de un .sch o un .kicad\_pcb).

En la mayoría de los conflictos se debe optar por una versión o la otra .

Si nos pasa que el GIT nos deja un archivo para “Mergear”, ignorarlo y solucionar el problema eligiendo una versión o la otra. Investigar “git merge --strategy ours/theirs”.

Este mismo problema lo tendremos con cualquier otro archivo binario que llevemos por GIT.

**Conclusión:** Más de una persona no puede trabajar al mismo tiempo sobre un archivo KiCad.

# Ejercicio 2 - Seguimiento con GIT

Ingresamos a GIT nuestros avances, con la versión casi terminada de nuestro esquemático.

- 1) Verificar ERC.
- 2) Generar netlist y el BOM.
- 3) Revisar BOM.
- 4) Verificar DRC.
- 5) Agregar los archivos a GIT.

**Tip:** En general los archivos que pueden generarse a partir de otros **NO** deben subirse a GIT. Sin embargo a veces generar estos archivos puede ser complicado. En esos casos se puede subirlos solamente a fines de documentar *releases* importantes.

**Ejemplo:** El pdf del esquemático resulta redundante, pero en una release se facilita la lectura del esquemático a cualquier interesado. Otros ejemplos son el BOM o el .net.

**AGREGAR**

<b>ej2.kicad_pcb:</b>	Diseño del PCB.
<b>ej2.sch:</b>	Hoja principal.
<b>Rs232.sch:</b>	Hoja secundaria.
<b>Conektor.sch:</b>	Hoja secundaria.
<b>ej2.pro:</b>	Proyecto.
<b>ej2-cache.lib:</b>	Una copia de los símbolos usados.

**AGREGAR DEPENDE**

<b>ej2.net:</b>	Agregar si el cambio impacta en el PCB.
<b>ej2.csv:</b>	Agregar en versiones importantes/finales.
<b>ej2.pdf:</b>	Esquemático en pdf.

**NO SUBIR**

<b>*.bak :</b>	Archivos de backup
<b>ej2.xml:</b>	(agregar al .gitignore)

## Directorios ejercicio2

<b>libs:</b>	biblioteca provista + creada.
<b>datasheets:</b>	Podemos colocar una copia del pdf.
<b>doc:</b>	Un texto comentando el ejercicio.
<b>ej2.pretty:</b>	Aún no hay nada aquí.
<b>ej2.3dshapes:</b>	Aún no hay nada aquí.
<b>tools:</b>	Scripts para el BOM.
<b>gerbers:</b>	Aún no hay nada aquí.

# GIT - Comandos más utilizados y para investigar

Los comandos GIT que pueden ser de utilidad para el curso son:

```
$ git config –global VARNAME VARVALUE  
$ git config –list  
$ git init  
$ git status -s  
$ git add ARCHIVO  
$ git commit -m “Descripción”  
$ git clone REMOTE-REPO  
$ git remote -v  
$ git fetch  
$ git diff origin master –name-status  
$ git merge  
$ git pull  
$ git push origin master  
$ git log --name-status  
$ git add .gitignore  
$ git tag NOMBRETAG  
$ git push origin NOMBRETAG  
$ git log –abbrev-commit  
$ git mv FILE NEWFILE  
$ git log –follow  
$ git rm FILE  
$ git log --name-status --diff-filter=D
```

Opciones para investigar:

```
$ git add -u  
$ git add -A  
$ git commit -a -m “Descripción”  
$ git checkout file.  
$ git merge --strategy ours/theirs  
$ git checkout --conflict=diff3 – file.txt  
$ git reset --hard  
$git clean
```

Opciones para manejo de ramas:

```
$ git checkout -b RAMA  
$ git checkout master  
$ git branch -d RAMA  
$ git log --graph  
$ git branch -a
```

En caso de conflictos nos pueden ayudar “git checkout” y “git reset” entre otros comandos.

# Autores e imágenes de esta presentación

Versión - 13/06/16

Autor de esta adaptación y contacto:  
Diego Brengi - [djavier@ieee.org](mailto:djavier@ieee.org)



**“Escuela KiCad - Curso de diseño de circuitos impresos”**  
*Preparado para la Carrera de Especialización en Sistemas Embebidos  
del LSE-FIUBA (CESE)*

Carátula principal:

Foto titulada “Circuit” de Yuri Samoilov bajo licencia CC-BY disponible en  
<https://www.flickr.com/photos/yusamoilov/14011462899/>

Fondo de la presentación:

Foto titulada “computer motherboard tracks” de Creativity103 bajo licencia CC-BY  
disponible en: [https://www.flickr.com/photos/creative\\_stock/5228433146/](https://www.flickr.com/photos/creative_stock/5228433146/)

Las imágenes de clipart se tomaron de: <https://openclipart.org/>

Los demás logos corresponden a proyectos de Software Libre u Open Source. Consultar  
cada licencia en particular.

Todas las capturas de pantalla fueron realizadas por los autores y están bajo la misma  
licencia que esta presentación.

Las imágenes del circuito esquemático con un BUS y la interconexión de tres BUSES  
fueron obtenidas del manual de KiCad <http://docs.kicad-pcb.org/en/eeschema.html> que  
pose licencia CC BY 3.0.

El resto de las imágenes se cita la fuente debajo de cada una.

