

# VHDL

Apuntes ejercicios

# Prácticas VHDL→ejercicios

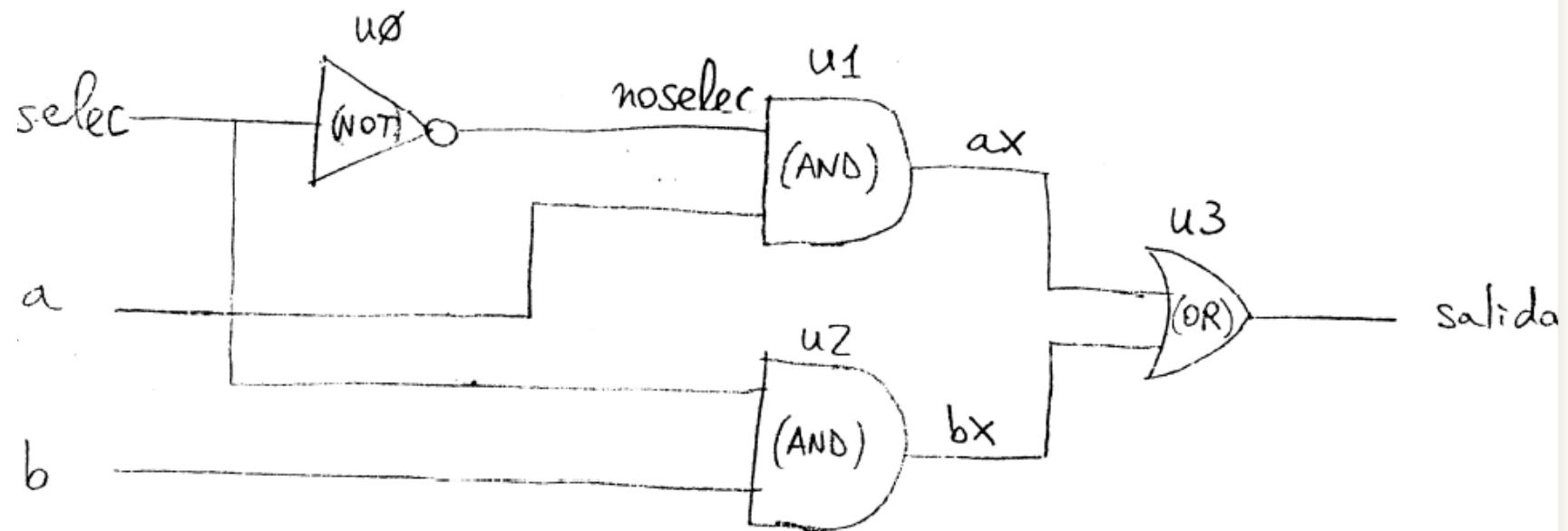
## Ejercicio 2

Describir en VHDL un circuito que multiplexe dos líneas (a y b) de un bit, a una sola línea (salida) también de un bit. La señal selec sirve para indicar que a la salida se tiene la línea a (selec = '0') o b (selec = '1').

# Prácticas VHDL → ejercicios

## Ejercicio 2

Circuito implementado con puertas lógicas que realiza la función de multiplexación:



# Prácticas VHDL → ejercicios

## Ejercicio 2

Pasos:

- 1º Definir (declarar) la entidad del circuito
- 2º Definir una o varias arquitecturas
- 3º Definir un “banco de pruebas”

# Prácticas VHDL → ejercicios

## Ejercicio 2

1º Definir (declarar) la entidad del  
circuito → definir sus entradas y salidas:

```
entity      mux is  
port (    a:      in   bit;  
          b:      in   bit;  
          selec:  in   bit;  
          salida: out  bit);  
end mux;
```

# Prácticas VHDL → ejercicios

## Ejercicio 2

1º Definir (declarar) la entidad del circuito

2º: Definir una o varias arquitecturas:

Especifican el funcionamiento del módulo definido en la entidad.

Corresponde a un tipo de descripción:

estructural

comportamental

de flujo de datos

# Prácticas VHDL → ejercicios

## Ejercicio 2

Tipos de descripción:

**Descr. Estructural:** Especificar componentes del sistema y sus interconexiones

**Descr. de flujo de datos:** Descripción comportamental con ejecución concurrente de las sentencias

**Descr. Comportamental:** Especifica la funcionalidad del sistema: entradas, salidas y correspondencia entre ambas. Pueden describirse partes con instrucciones secuenciales

# Prácticas VHDL → ejercicios


## Ejercicio 2

Tipos de descripción:

**Descr. Estructural**

**Descr. de flujo de datos**

**Descr. Comportamental**



Mayor nivel de abstracción





## Ejercicio 2. 2º: Definir una o varias arquitecturas:

- Descripción comportamental:  
--Arquitectura comportamental de mux  
begin  
  process (a, b, selec)  
  begin  
    if (selec = `0`)then  
      salida <= a;  
    else  
      salida <= b;  
    end if;  
  end process;  
end comportamental;



Las instrucciones del proceso se ejecutan cada vez que alguna de las señales de la lista sensible cambia

## **Ejercicio 2. 2º: Definir una o varias arquitecturas:**

- Descripción estructural de mux:  
--Arquitectura estructural de mux  
architecture estructural of mux is  
component and2  
port (e1,e2: in bit; y:out bit);  
end component;

# Prácticas VHDL→ejerciciosEjercicio 2(cont)

Descripción estructural de mux:

architecture estructural of mux is

  component and2

    port (e1,e2: in bit; y:out bit);

  end component;

  component or2

    port (e1,e2: in bit; y:out bit);

  end component;

  component inv

    port (e: in bit; y:out bit);

  end component;

# Prácticas VHDL → ejercicios Ejercicio 2(cont)

Descripción estructural de mux:

Pasos:

1º definir los **componentes**: indicar sus entradas y salidas con **port**.

Los componentes deben tener una entidad y una arquitectura propia indicando su funcionamiento.

Supondremos que ya están definidos.

(Normalmente, en una librería separada).

2º

# Prácticas VHDL→ejerciciosEjercicio 2(cont)

Descripción estructural de mux:

Pasos:

1º los componentes: entradas y salidas con port.

2º Declarar como señales las conexiones entre componentes, nombrándolas.

# Prácticas VHDL → ejercicios Ejercicio 2(cont)

Descripción estructural de mux:

Pasos:

1º los **componentes**: entradas y salidas con **port**.

2º Declarar **señales**

3º Instanciación:

- .-Asignar a cada componente un nombre concreto
- .-Definir su tipo de componente
- .-Asignar las conexiones → interconectarán a los componentes entre sí.

# Prácticas VHDL→ejercicios

## Descripción estructural Ejercicio 2

- 1º Definir (declarar) la entidad del circuito
- 2º Definir una o varias arquitecturas
- 3º Definir un banco de pruebas para realizar la simulación



# Prácticas VHDL→ejercicios

## Descripción estructural Ejercicio 2

3º Definir un banco de pruebas para realizar la simulación

--Banco de pruebas de la entidad mux

```
entity t_mux is --no tiene entradas ni salidas
```

```
end t_mux;
```

```
architecture estimulos of t_mux is
```

```
signal t_a,t_b,t_selec,t_salida: bit);
```

# Prácticas VHDL→ejercicios

## Descripción estructural Ejercicio 2

3º Definir un banco de pruebas para realizar la simulación

--Banco de pruebas de la entidad mux

```
entity t_mux is --no tiene entradas ni salidas
```

```
end t_mux;
```

```
architecture estimulos of t_mux is
```

```
signal t_a,t_b,t_selec,t_salida: bit);
```

# Prácticas VHDL→ejercicios

## Descripción estructural Ejercicio 2

3º--Banco de pruebas de la entidad mux

```
component mux
```

```
    port (a,b,selec: in bit; salida: out bit);
```

```
end component;
```

```
begin
```

```
mux1: mux port map (t_a,t_b,t_selec,t_salida);
```

```
process
```

```
begin
```

```
    t_a<='0';
```

```
    t_b<='1';
```

```
    t_selec<='0';
```

```
    wait for 20 ns;
```

# Prácticas VHDL→ejercicios

## Descripción estructural Ejercicio 2

3º--Banco de pruebas de la entidad mux

```
wait for 20 ns;
```

```
t_selec<='1';
```

```
wait for 20 ns;
```

```
t_a<='1';
```

```
t_b<='0';
```

```
wait for 20 ns;
```

# Prácticas VHDL→ejercicios

## Descripción estructural **Ejercicio 2**

3º--Banco de pruebas de la entidad mux

wait for 20 ns;

t\_selec<='0';

wait for 20ns;

wait;

end process;

end estimulos;

# Prácticas VHDL→ejercicios

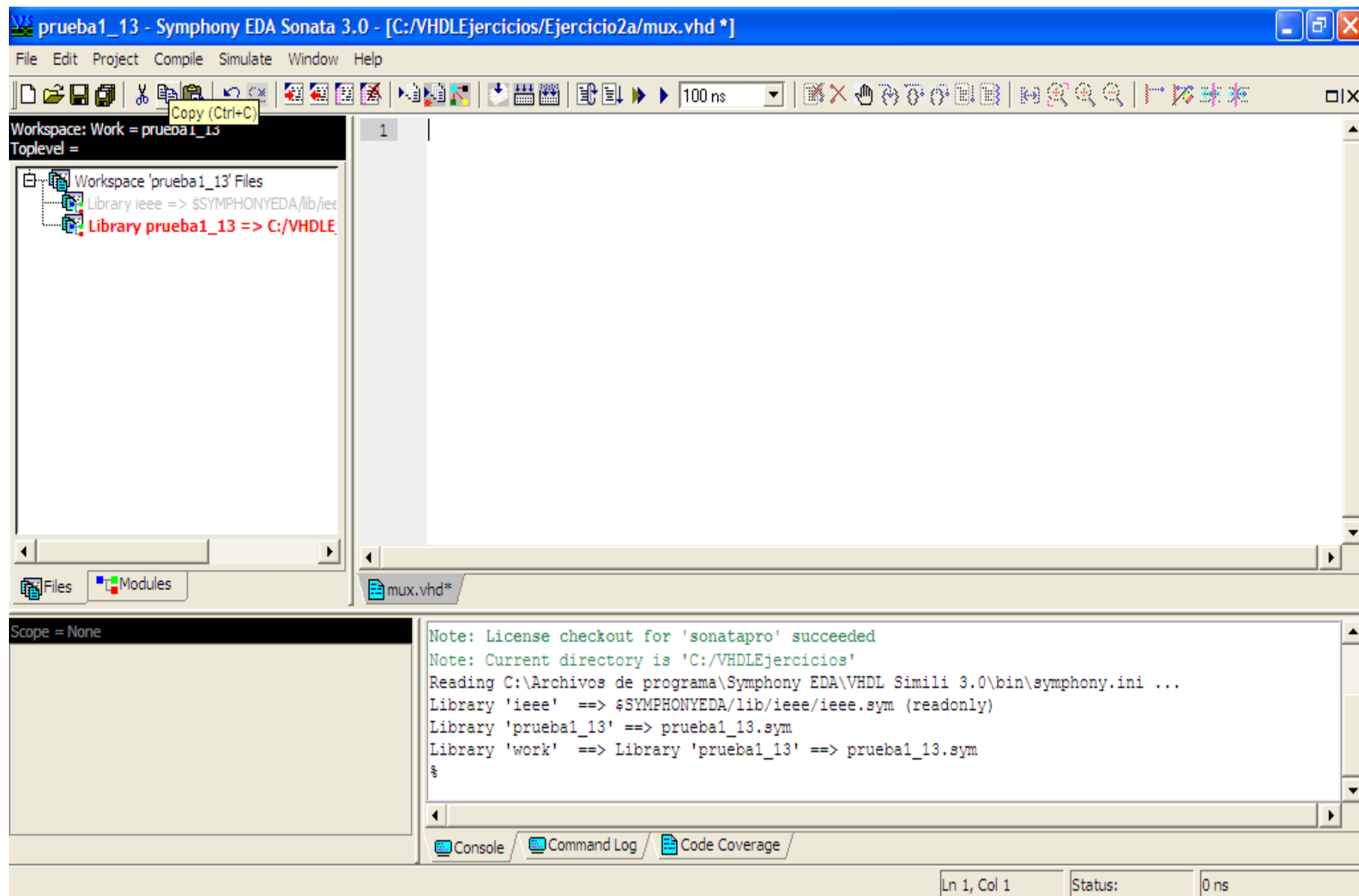
## Descripción estructural **Ejercicio 2.** Bancos de pruebas

- Programas para comprobar el funcionamiento de un modelo VHDL
- Compuesto por:
  - Entidad
  - Arquitectura:
    - Una instancia del componente a probar
    - Procesos para generar secuencias de valores a las señales conectadas a la instancia del componente
    - Procesos para comprobar las señales de salida o bien se comprueban en el simulador

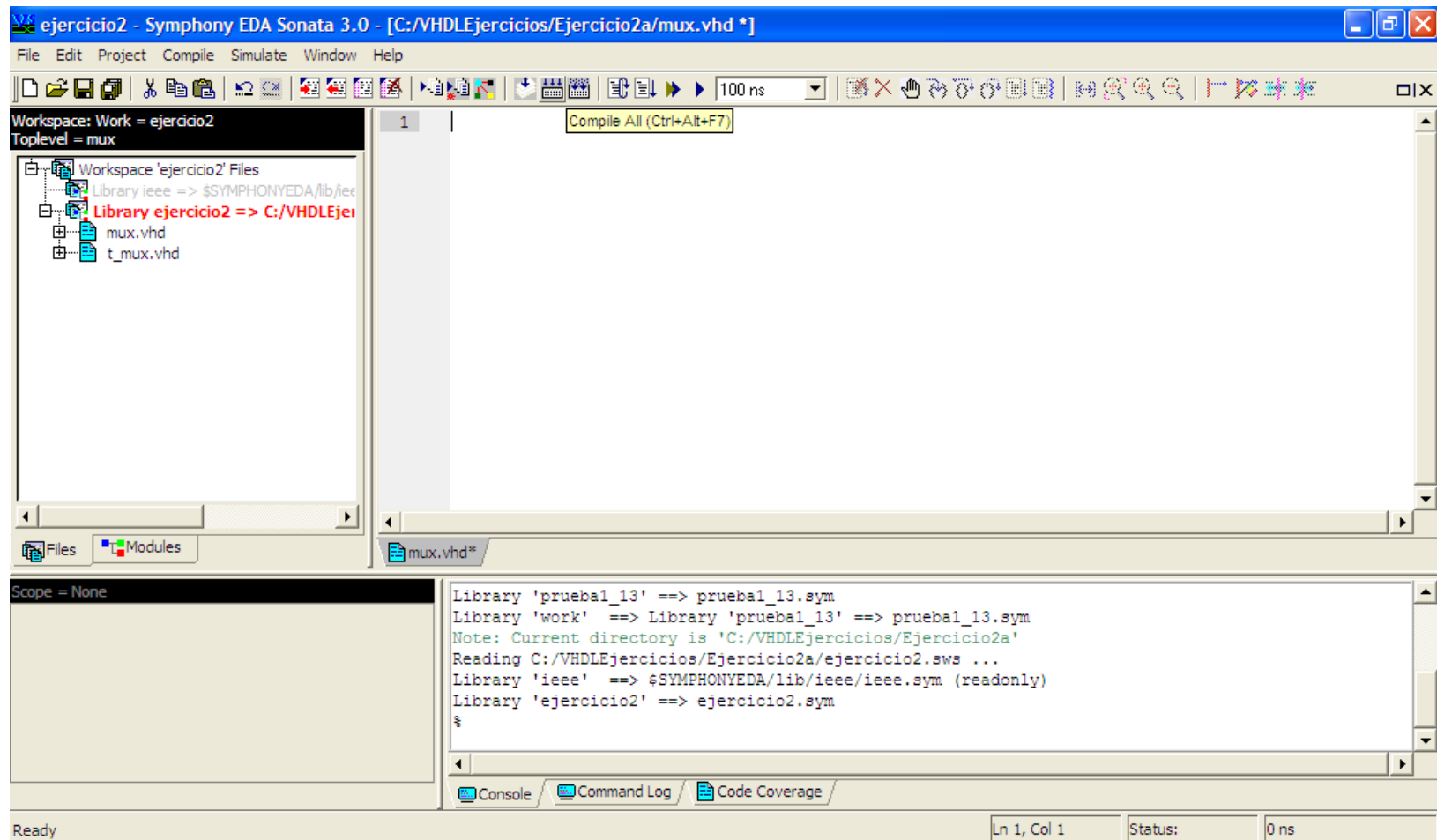
# Prácticas VHDL→ejercicios

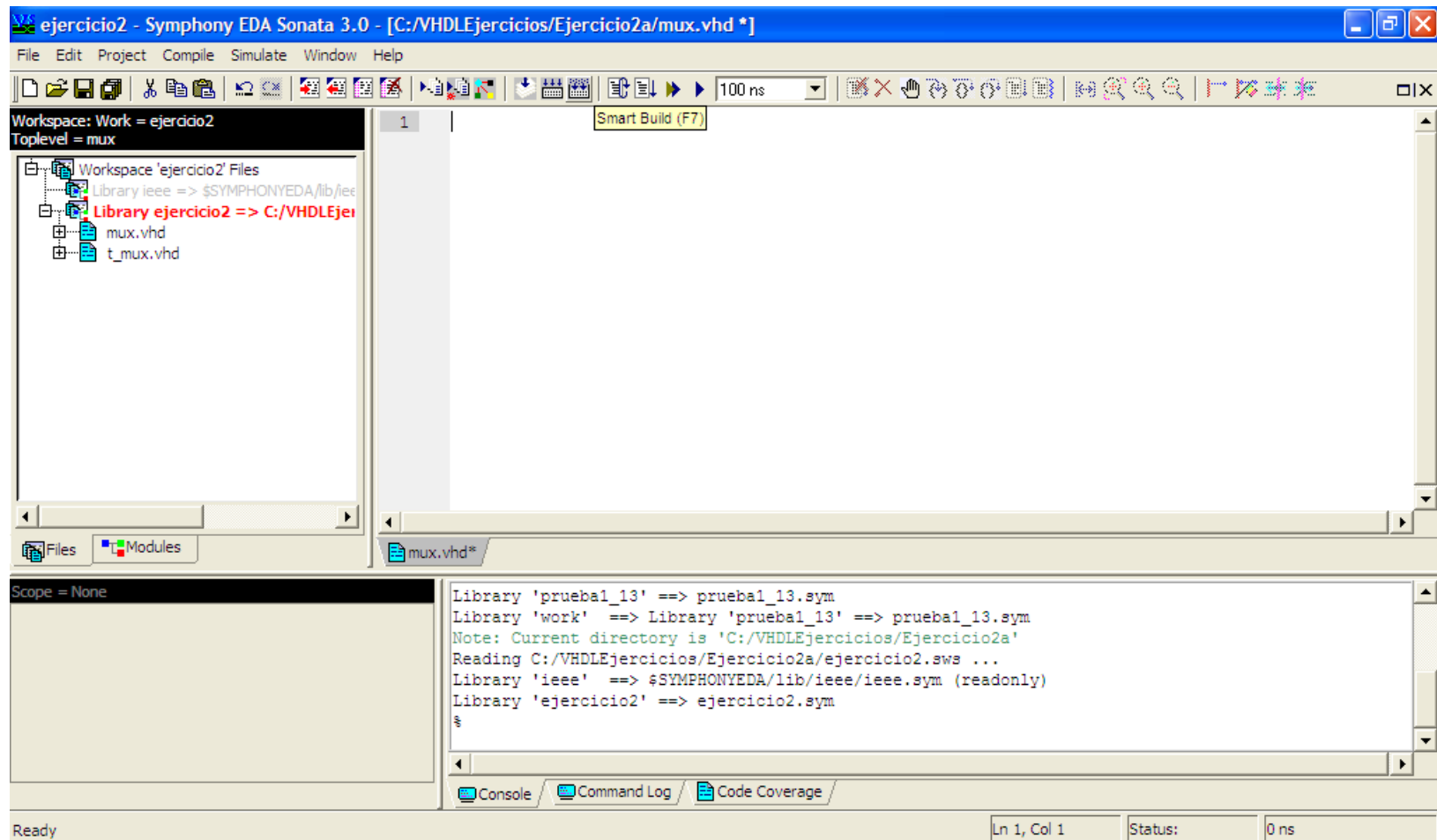
## Descripción estructural **Ejercicio 2. simulación**

- Abrir sonata
- Crear nuevo workspace
- Crear los archivos con el código fuente:
  - File → New
  - .....
  - save as→nombre (mux.vhd t\_mux.vhd)
- Añadir archivos al proyecto:
  - Project→add files









## **Prácticas VHDL→ejercicios**

### **Descripción estructural Ejercicio 2. simulación**

- (mux.vhd t\_mux.vhd)
- Compilar archivos: compile all y Smart build

## **Prácticas VHDL→ejercicios**

### **Descripción estructural Ejercicio 2. simulación**

- Ejecutar el simulador: Build and restart simulation
  - Seleccionar banco de pruebas t\_mux como nivel superior (da igual entidad o arquitectura)
  - Seleccionar todas las señales (botón derecho, select all)
  - Arrastrarlas a signals
  - Correr la simulación

# Prácticas VHDL → ejercicios

## Descripción estructural Ejercicio 2. simulación

The screenshot displays the VHDL simulation environment, specifically the waveform viewer and console output. The top window shows the file 'vhd\_001-027\_ApuntosEjercicios.pdf' in Adobe Reader. Below it, the 'ejercicio2 - Symphony EDA Sonata 3.0' window is open, showing the workspace with the top-level entity 't\_mux'. The waveform viewer displays the signals 't\_a', 't\_b', 't\_selec', and 't\_salida' over time, with a cursor at 0ps. The console output shows the design load/elaboration time and the command to add waveforms.

Workspace: Work = ejercicio2  
Toplevel = t\_mux

Signals

Signal	Value
t_a	'0'
t_b	'0'
t_selec	'0'
t_salida	'0'

Source: Simulator  
Cursor: 0ps

Scope = :t\_mux:

Signals

Signal	Value
t_a	'0'
t_b	'0'
t_selec	'0'
t_salida	'0'

Reading ejercicio2.sym\mux\\_comportamental.var

# of Signals = 8  
# of Components = 1  
# of Processes = 2

Design Load/Elaboration Elapsed Time: 00h:00m:00s:047ms

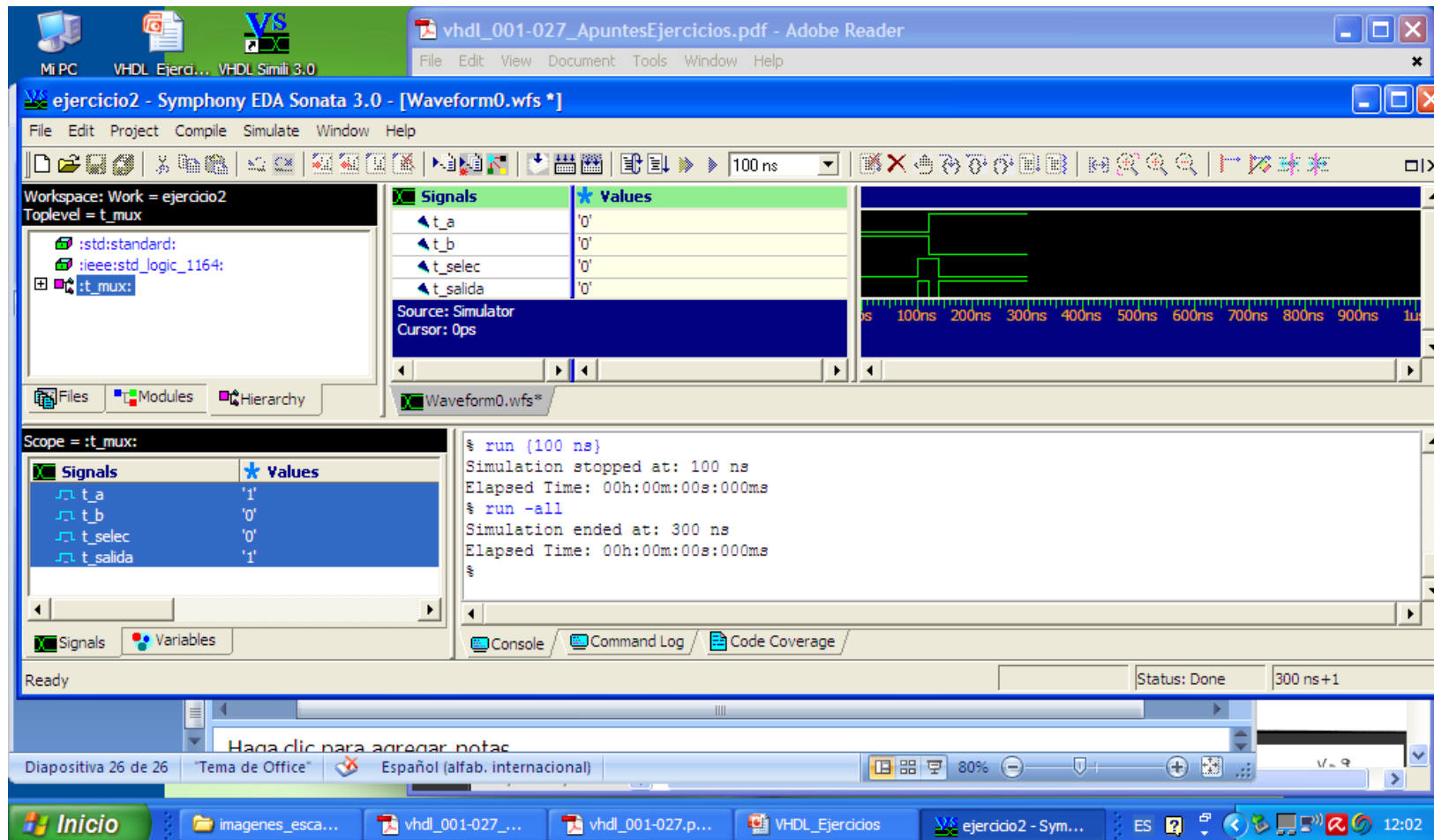
% add wave :t\_mux:t\_a :t\_mux:t\_b :t\_mux:t\_selec :t\_mux:t\_salida

Status: Stopped 0 ps+0

Diapositiva 26 de 26 Tema de Office Español (alfab. internacional) 80% 12:02

# Prácticas VHDL → ejercicios

## Descripción estructural Ejercicio 2. simulación



## **Prácticas VHDL→ejercicios**

### **Descripción estructural Ejercicio 2. simulación**

- Leer los valores de las señales picando con el ratón
- Finalizar la simulación

# Descripción estructural

Workspace: Work = ejercicio2  
Toplevel = t\_mux

- std:standard:
- ieee:std\_logic\_1164:
- t\_mux:

Signals	Values
t_a	'0'
t_b	'0'
t_selec	'0'
t_salida	'0'

Source: Simulator  
Cursor: 0ps

End Simulation

111111.111ps

Scope = :t\_mux:

Signals	Values
t_a	'1'
t_b	'0'
t_selec	'0'
t_salida	'1'

```
% run
Simulation stopped at: 100 ns
Elapsed Time: 00h:00m:00s:000ms
% run {100 ns}
Simulation stopped at: 200 ns
Elapsed Time: 00h:00m:00s:000ms
%
```

Status: Stopped 200 ns+0

Diapositiva 29 de 29 "Tema de Office" Español (alfab. internacional)

Inicio imagenes\_esca... vhd\_001-027... vhd\_001-027.p... VHDL\_Ejercicios ejercicio2 - Sym...



# Prácticas VHDL → ejercicios

## Ejercicio 2

1º Definir (declarar) la entidad del circuito

2º: Definir una o varias arquitecturas:

Especifican el funcionamiento del módulo definido en la entidad.

Corresponde a un tipo de descripción:

estructural

comportamental

de flujo de datos

# Prácticas VHDL→ejercicios

## Descripción de flujo de datos

- Instrucciones que se ejecutan concurrentemente:  
Permiten ejecución paralela
- Las sentencias indican asignaciones a señales, conexiones, es como si fuera una única instrucción que se ejecuta continuamente.
- Es más fácil sintetizar un circuito descrito en flujo de datos (RTL) que con mayor nivel de abstracción, ya que se corresponde más directamente con su implementación hardware que en una descripción secuencial