

	A	B	C	D	E
	Nombre del Proyecto	Función	Módulos	Concepto introducido	Sentencias
1	Project_01	Puerta And de dos entradas	AND_GATE	Operador de asignación	Concurrente
2	Project_02	Puerta NOR de dos entradas de dos bits	NOR_GATE	1. Arrays (STD_LOGIC_VECTOR) 2. XDC (GUI)	Concurrente
3	Project_03	Puerta XOR parametrizada	XOR_GATE	Generics	Concurrente
4	Project_04	1. Multiplexor de dos entradas, asignación condicional CONCURRENTE 2. Multiplexor genérico, asignación condicional SECUENCIAL	MUX2_1bit MUX2	1.- Dos ficheros VHD en un proyecto: Set as TOP 2.- WHEN-ELSE 3.- Process 4.- IF-ELSE-END	Concurrente
5	Project_05	Decodificador de 2 a 4	DEC_2to4	WHEN-ELSE	Concurrente
6	Project_06	Full adder (1-bit)	FULL_ADDER	Estilo DATAFLOW (Poco interesante)	Concurrente
7	Project_07	Multiplexor de cuatro entradas (genérico)	MUX4	WITH-SELECT	Concurrente
8	Project_08	Convertidor de binario (4 bits) a 7 segmentos con selección de display	DISP7SEG	1. WITH-SELECT 2.- Concurrence	Concurrente
9	Project_09	SHOW_DATA	SHOW_DATA (TOP) DISP7SEG MUX2 (WIDTH=4)	1.- Descripción estructural: a. Instancia de componentes genericos pasando un valor (Generic Map) b. Asignar valor fijo a un puerto de entrada de algún componente 2.- Dummy signal	Concurrente
10	Project_10	1 bit ALU: AND, NOR, MOV, Full ADD. Estructural	ALU_1bit	1.- Descripción estructural: componentes, señales y mapeado 2.- Posibilidad de no añadir modulo MUX y describirlo en el fichero TOP (WITH-SELECT) 3.- Modificar módulos solo de forma local	Concurrente
11	Project_11	ALU genérica. Estilo comportamental (Concurrente).	ALU_0_N_bits	1.- ALU simple Sentencias concurrentes (WITH-SELECT) 2.- Operaciones Aritméticas: suma y resta (use IEEE.NUMERIC_STD.ALL, +,-) 3.- Conversión tipos STD_LOGIC_VECTOR a SIGNED y UNSIGNED 4.- OTHERS 5.- Uso de constante ZERO	Concurrente
12	Project_11	ALU genérica. Estilo comportamental. (Secuencial)	ALU_1_N_bits	1.- ALU con sentencias secuenciales (PROCESS, CASE) 2.- Operaciones de desplazamiento lógico (SRL, SLL) 3.- Ampliar tamaño vectores (&)	Secuencial
13	Project_12	Decodificador genérico n to 2**n	DECO_N_BITS	1.- Uso del operador potencia (**) 2.- Uso de desplazamiento lógico 3.- Asignación selectiva de valor a una constante (0=>'1', others=>'0')	Secuencial
14	Project_13	Flip-Flop tipo D	FFD_Basic	1.- Añadir CLK a testbench	
15	Project_13	Flip-Flop tipo D	FFD	2.- Añadir CLK a fichero de restricciones	
16	Project_14	Conmutador	TOOGLE_00	1.- Ver señales internas en la simulación	
17	Project_14	Conmutador	TOOGLE_01	2.- Integrar dos módulos en un solo proceso 3.- Sincronizar entradas, detectar flancos en entradas 4.- Descripción comportamental en un solo fichero versus estructural	
18	Project_15	Detector de flancos positivos	EDGE_DETECTOR	1.- Filtrar la duración de una entrada a un solo ciclo de reloj. (ENABLE de duración 1 ciclo) 2.- Registrar entradas para garantizar que duren siempre un ciclo de reloj completo	
19	Project_16	Registro de N bits	REG_N_bits	1.- Estructura para almacenar más de 1 bit, valores con un mayor número de bits	
20	Project_17	Contador de N bits	COUNTER_N_bits		Comportamental
21	Project_18	Su turno: incrementa una cuenta de 0 a 9 cada vez que se actúa sobre un pulsador (Enable)	YOUR_TURN COUNTER_N_Bits DISP7SEG EDGE_DETECTOR		Estructural