

Importar as configurações dos pinos salvas em um arquivo .csv

- O resultado do procedimento de importação apresentado aqui está no arquivo **Output_Pin_Planner.pdf**.

1 - Arquivo com as configurações dos pinos do kit FPGA:

The screenshot shows a CSV file titled "Config_Pinos_SemVGA.csv" in a text editor. The file contains 108 lines of configuration data, each line starting with a line number (e.g., 25, 26, 27, ..., 56) followed by a pin description and its settings. The data includes various pins like DIG[1], DIG[0], FPGA_BUZZER, FPGA_CLK, etc., with their respective output types, pins, and voltages. The file is located at C:\Arduino_RPi_IoT\FPGA\FPGA_Cyclone\KitEasyFPGA-Dualv1m\Procedimentos_Quartus\Importar_Confs_Pinos\Config_Pinos_SemVGA.csv.

Line	Pin Description	Output Type	Pin Number	Voltage
25	DIG[1]	Output	PIN_135	8, B8_N0, PIN_135, 2.5 V,,
26	DIG[0]	Output	PIN_133	8, B8_N0, PIN_133, 2.5 V,,
27	FPGA_BUZZER	Output	PIN_110	7, B7_N0, PIN_110, 2.5 V,,
28	FPGA_CLK	Input	PIN_23	1, B1_N0, PIN_23, 2.5 V,,
29	FPGA_RST	Input	PIN_25	2, B2_N0, PIN_25, 2.5 V,,
30	FPGA_SCL	Output	PIN_112	7, B7_N0, PIN_112, 2.5 V,,
31	FPGA_SDA	Bidir	PIN_113	7, B7_N0, PIN_113, 2.5 V,,
32	I2C_SCL	Output	PIN_99	6, B6_N0, PIN_99, 2.5 V,,
33	I2C_SDA	Bidir	PIN_98	6, B6_N0, PIN_98, 2.5 V,,
34	IR	Input	PIN_100	6, B6_N0, PIN_100, 2.5 V,,
35	KEY[3]	Input	PIN_91	6, B6_N0, PIN_91, 2.5 V,,
36	KEY[2]	Input	PIN_90	6, B6_N0, PIN_90, 2.5 V,,
37	KEY[1]	Input	PIN_89	5, B5_N0, PIN_89, 2.5 V,,
38	KEY[0]	Input	PIN_88	5, B5_N0, PIN_88, 2.5 V,,
39	LCD_D[7]	Output	PIN_11	1, B1_N0, PIN_11, 2.5 V,,
40	LCD_D[6]	Output	PIN_7	1, B1_N0, PIN_7, 2.5 V,,
41	LCD_D[5]	Output	PIN_10	1, B1_N0, PIN_10, 2.5 V,,
42	LCD_D[4]	Output	PIN_2	1, B1_N0, PIN_2, 2.5 V,,
43	LCD_D[3]	Output	PIN_3	1, B1_N0, PIN_3, 2.5 V,,
44	LCD_D[2]	Output	PIN_144	8, B8_N0, PIN_144, 2.5 V,,
45	LCD_D[1]	Output	PIN_1	1, B1_N0, PIN_1, 2.5 V,,
46	LCD_D[0]	Output	PIN_142	8, B8_N0, PIN_142, 2.5 V,,
47	LCD_E	Output	PIN_143	8, B8_N0, PIN_143, 2.5 V,,
48	LCD_RS	Output	PIN_141	8, B8_N0, PIN_141, 2.5 V,,
49	LCD_RW	Output	PIN_138	8, B8_N0, PIN_138, 2.5 V,,
50	LDQM	Output	PIN_42	3, B3_N0, PIN_42, 2.5 V,,
51	LED[3]	Output	PIN_84	5, B5_N0, PIN_84, 2.5 V,,
52	LED[2]	Output	PIN_85	5, B5_N0, PIN_85, 2.5 V,,
53	LED[1]	Output	PIN_86	5, B5_N0, PIN_86, 2.5 V,,
54	LED[0]	Output	PIN_87	5, B5_N0, PIN_87, 2.5 V,,
55	PS_CLOCK	Input	PIN_119	7, B7_N0, PIN_119, 2.5 V,,
56	PS_DATA	Input	PIN_120	7, B7_N0, PIN_120, 2.5 V,,

Figure 1: Arquivo com as configurações dos pinos

- O arquivo **Config_Pinos_SemVGA.csv** contém as configurações dos pinos do kit RZ-EasyFPGA A2.2.
- Esse arquivo será importado ao projeto do Quartus.

2 - Importar as configurações do arquivo .csv no projeto

3 - Visualizando o resultado da importação no arquivo Teste.qsf

4 - Programa de teste:

```
module Teste(
```

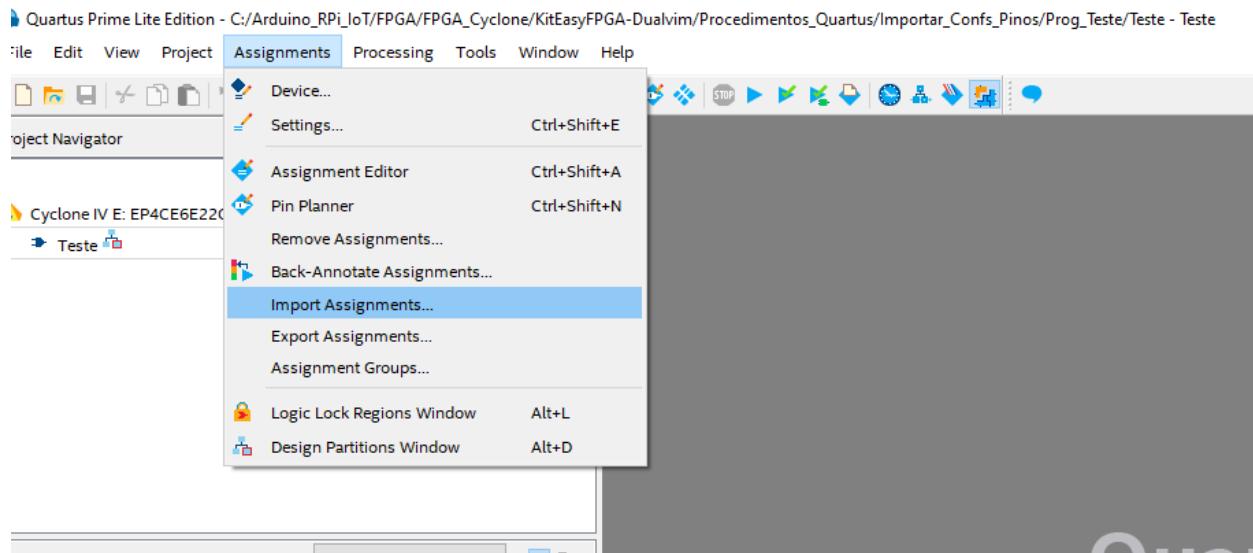


Figure 2: Assignments/Import Assignments...

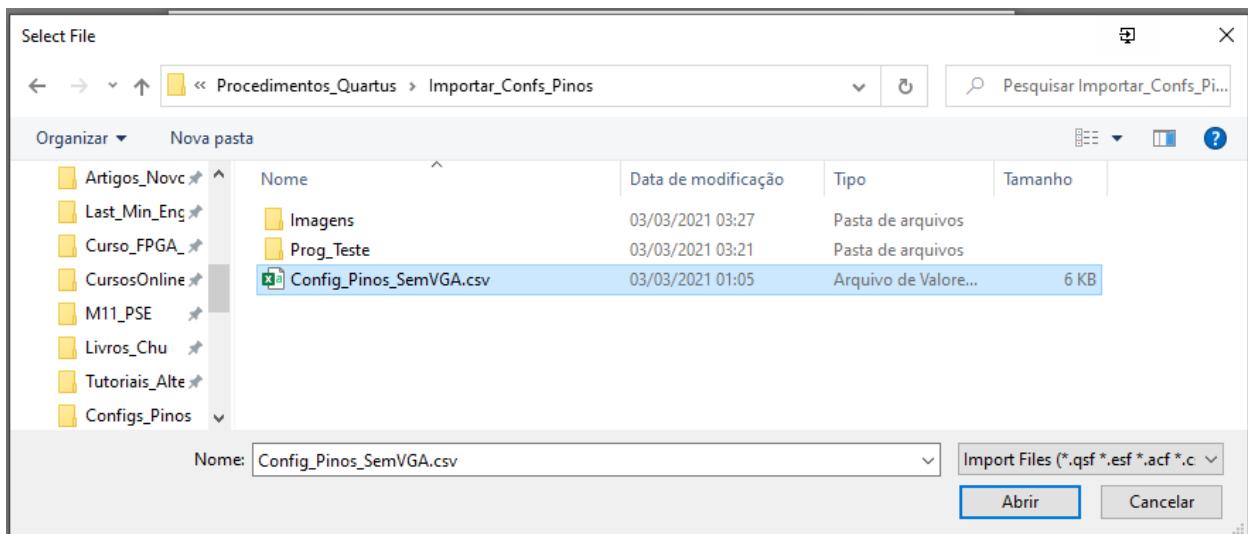


Figure 3: Selezionando o arquivo com as configurações

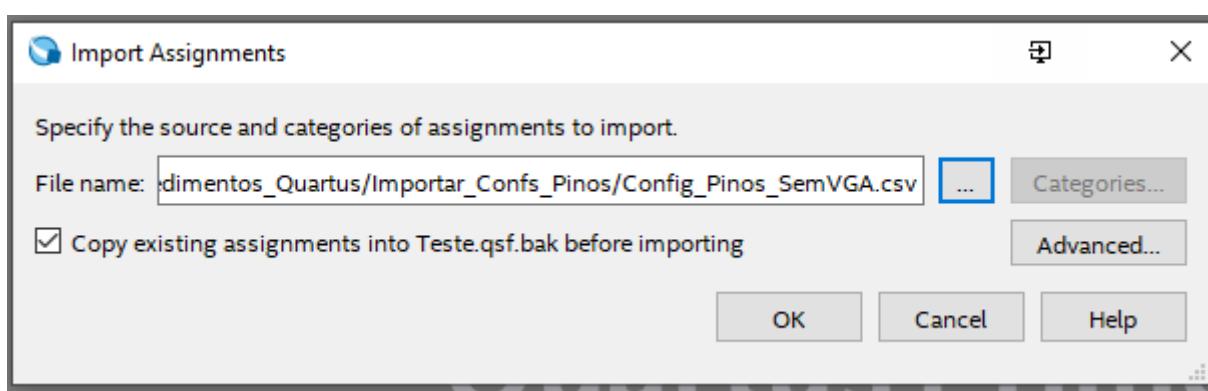
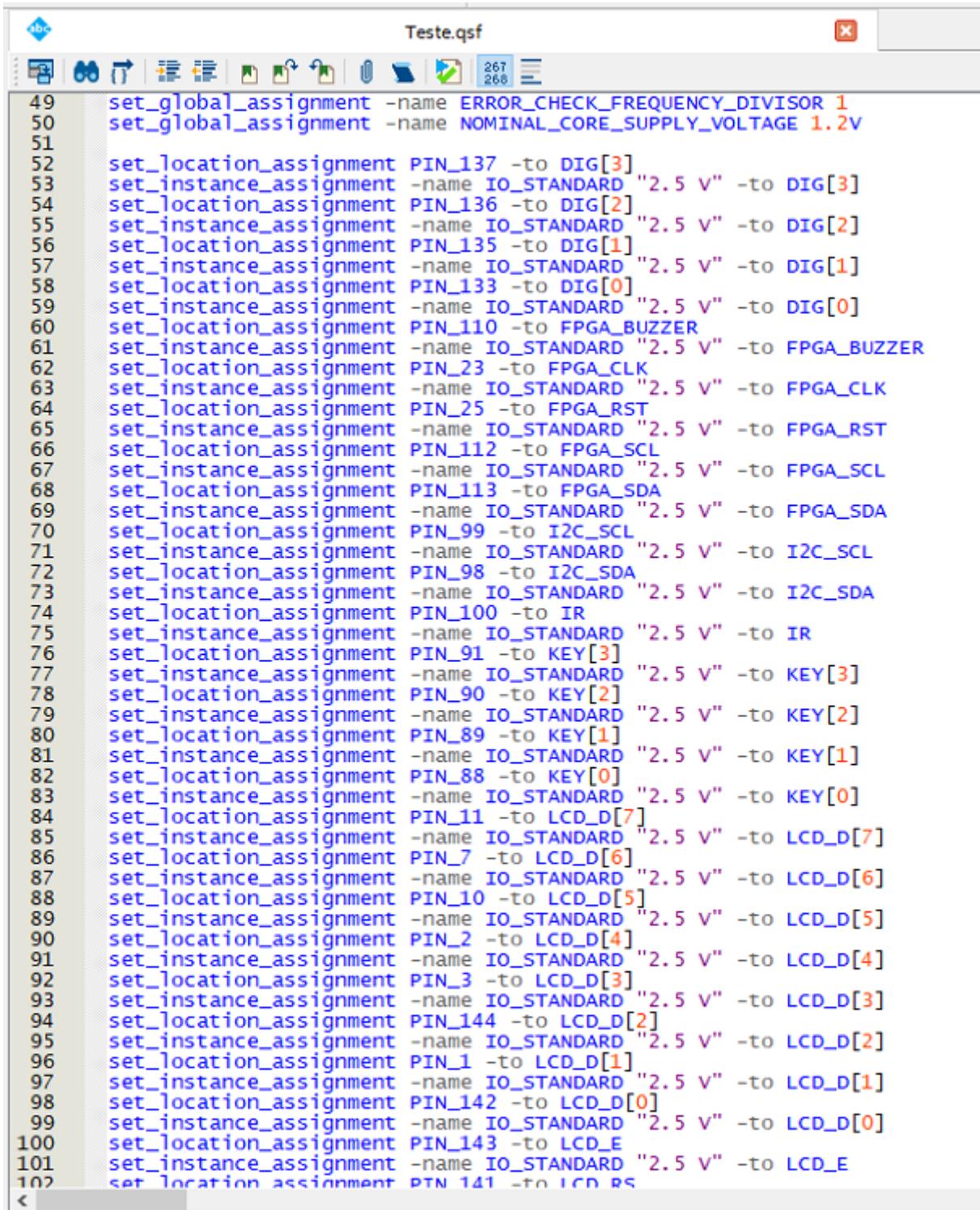


Figure 4: Confirmar a importação



The screenshot shows a software interface with a title bar 'abc' and 'Teste.qsf'. The main area is a code editor with the following content:

```
49 set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 1
50 set_global_assignment -name NOMINAL_CORE_SUPPLY_VOLTAGE 1.2V
51
52 set_location_assignment PIN_137 -to DIG[3]
53 set_instance_assignment -name IO_STANDARD "2.5 V" -to DIG[3]
54 set_location_assignment PIN_136 -to DIG[2]
55 set_instance_assignment -name IO_STANDARD "2.5 V" -to DIG[2]
56 set_location_assignment PIN_135 -to DIG[1]
57 set_instance_assignment -name IO_STANDARD "2.5 V" -to DIG[1]
58 set_location_assignment PIN_133 -to DIG[0]
59 set_instance_assignment -name IO_STANDARD "2.5 V" -to DIG[0]
60 set_location_assignment PIN_110 -to FPGA_BUZZER
61 set_instance_assignment -name IO_STANDARD "2.5 V" -to FPGA_BUZZER
62 set_location_assignment PIN_23 -to FPGA_CLK
63 set_instance_assignment -name IO_STANDARD "2.5 V" -to FPGA_CLK
64 set_location_assignment PIN_25 -to FPGA_RST
65 set_instance_assignment -name IO_STANDARD "2.5 V" -to FPGA_RST
66 set_location_assignment PIN_112 -to FPGA_SCL
67 set_instance_assignment -name IO_STANDARD "2.5 V" -to FPGA_SCL
68 set_location_assignment PIN_113 -to FPGA_SDA
69 set_instance_assignment -name IO_STANDARD "2.5 V" -to FPGA_SDA
70 set_location_assignment PIN_99 -to I2C_SCL
71 set_instance_assignment -name IO_STANDARD "2.5 V" -to I2C_SCL
72 set_location_assignment PIN_98 -to I2C_SDA
73 set_instance_assignment -name IO_STANDARD "2.5 V" -to I2C_SDA
74 set_location_assignment PIN_100 -to IR
75 set_instance_assignment -name IO_STANDARD "2.5 V" -to IR
76 set_location_assignment PIN_91 -to KEY[3]
77 set_instance_assignment -name IO_STANDARD "2.5 V" -to KEY[3]
78 set_location_assignment PIN_90 -to KEY[2]
79 set_instance_assignment -name IO_STANDARD "2.5 V" -to KEY[2]
80 set_location_assignment PIN_89 -to KEY[1]
81 set_instance_assignment -name IO_STANDARD "2.5 V" -to KEY[1]
82 set_location_assignment PIN_88 -to KEY[0]
83 set_instance_assignment -name IO_STANDARD "2.5 V" -to KEY[0]
84 set_location_assignment PIN_11 -to LCD_D[7]
85 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[7]
86 set_location_assignment PIN_7 -to LCD_D[6]
87 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[6]
88 set_location_assignment PIN_10 -to LCD_D[5]
89 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[5]
90 set_location_assignment PIN_2 -to LCD_D[4]
91 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[4]
92 set_location_assignment PIN_3 -to LCD_D[3]
93 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[3]
94 set_location_assignment PIN_144 -to LCD_D[2]
95 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[2]
96 set_location_assignment PIN_1 -to LCD_D[1]
97 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[1]
98 set_location_assignment PIN_142 -to LCD_D[0]
99 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_D[0]
100 set_location_assignment PIN_143 -to LCD_E
101 set_instance_assignment -name IO_STANDARD "2.5 V" -to LCD_E
102 set_location_assignment PTN_141 -to LCD_RS
```

Figure 5: Resultado da importação no arquivo Teste.qsf

The screenshot shows a Verilog code editor window titled "Teste.v". The code is a module named "Teste" with various interface ports. The ports are grouped by comments and include:

- Clocks:** FPGA_RST, FPGA_CLK
- Botoes:** KEY [3:0]
- LEDs:** LED [3:0]
- Display de 7 segmentos:** DIG [3:0], SEG [7:0]
- SDRAM:** SDRAM_CLK, SDRAM_CKE, SDRAM_A [11:0], SDRAM_BS [1:0], SDRAM_DQ [15:0], LDQM, UDQM, SDRAM_CS, SDRAM_WE, SDRAM_CAS, SDRAM_RAS
- LCD:** LCD_RS, LCD_RW, LCD_E, LCD_D [7:0]
- UART:** UART_RXD, UART_TXD
- PS/2:** PS_CLOCK, PS_DATA
- Interface I2C:** I2C_SCL, I2C_SDA, FPGA_SCL, FPGA_SDA
- Outros:** IR, FPGA_BT727FR

```
7  module Teste(
8      //////////////// clocks ///////////
9      input          FPGA_RST,
10     input          FPGA_CLK,
11
12     //////////////// Botoes ///////////
13     input [3:0]    KEY,
14
15
16     //////////////// LEDs ///////////
17     output [3:0]   LED,
18
19
20     //////////////// Display de 7 segmentos///////////
21     output [3:0]   DIG,
22     output [7:0]   SEG,
23
24     //////////////// SDRAM ///////////
25     output         SDRAM_CLK,
26     output         SDRAM_CKE,
27     output [11:0]  SDRAM_A,
28     output [1:0]   SDRAM_BS,
29     inout [15:0]  SDRAM_DQ,
30     output         LDQM,
31     output         UDQM,
32     output         SDRAM_CS,
33     output         SDRAM_WE,
34     output         SDRAM_CAS,
35     output         SDRAM_RAS,
36
37     //////////////// LCD ///////////
38     output         LCD_RS,
39     output         LCD_RW,
40     output         LCD_E,
41     output [7:0]  LCD_D,
42
43
44     //////////////// UART ///////////
45     input          UART_RXD,
46     output         UART_TXD,
47
48     //////////////// PS/2 ///////////
49     input          PS_CLOCK,
50     input          PS_DATA,
51
52     //////////////// Interface I2C ///////////
53     output         I2C_SCL,
54     inout          I2C_SDA,
55     output         FPGA_SCL,
56     inout          FPGA_SDA,
57
58     //////////////// Outros ///////////
59     input          IR,
60     inout         FPGA_BT727FR
```

Figure 6: Script em Verilog Teste.v

```

////////// Clocks //////////
input           FPGA_RST,
input           FPGA_CLK,
////////// Botões //////////
input [3: 0]    KEY,
////////// LEDs //////////
output [3: 0]   LED,
////////// Display de 7 segmentos/////////
output [3: 0]   DIG,
output [7: 0]   SEG,
////////// SDRAM //////////
output          SDRAM_CLK,
output          SDRAM_CKE,
output [11: 0]  SDRAM_A,
output [ 1: 0]  SDRAM_BS,
inout [15: 0]  SDRAM_DQ,
output          LDQM,
output          UDQM,
output          SDRAM_CS,
output          SDRAM_WE,
output          SDRAM_CAS,
output          SDRAM_RAS,
////////// LCD //////////
output          LCD_RS,
output          LCD_RW,
output          LCD_E,
output [7: 0]   LCD_D,
////////// UART //////////
input           UART_RXD,
output          UART_TXD,
////////// PS/2 //////////
input           PS_CLOCK,
input           PS_DATA,
////////// Interface I2C //////////
output          I2C_SCL,
inout          I2C_SDA,
output          FPGA_SCL,
inout          FPGA_SDA,
////////// Outros //////////
input           IR,
output          FPGA_BUZZER

);

//Acender os LEDs de acordo com o botão pressionado
assign LED[0] = ~KEY[0];
assign LED[1] = ~KEY[1];
assign LED[2] = ~KEY[2];
assign LED[3] = ~KEY[3];
endmodule

```

configurações importadas no *Pin Planner*

Flow Summary	
<<Filter>>	
Flow Status	Successful - Wed Mar 03 03:42:42 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	Teste
Top-level Entity Name	Teste
Family	Cyclone IV E
Device	EP4CE6E22C8
Timing Models	Final
Total logic elements	0 / 6,272 (0 %)
Total registers	0
Total pins	81 / 92 (88 %)
Total virtual pins	0
Total memory bits	0 / 276,480 (0 %)
Embedded Multiplier 9-bit elements	0 / 30 (0 %)
Total PLLs	0 / 2 (0 %)

Figure 7: *Compilation Report*

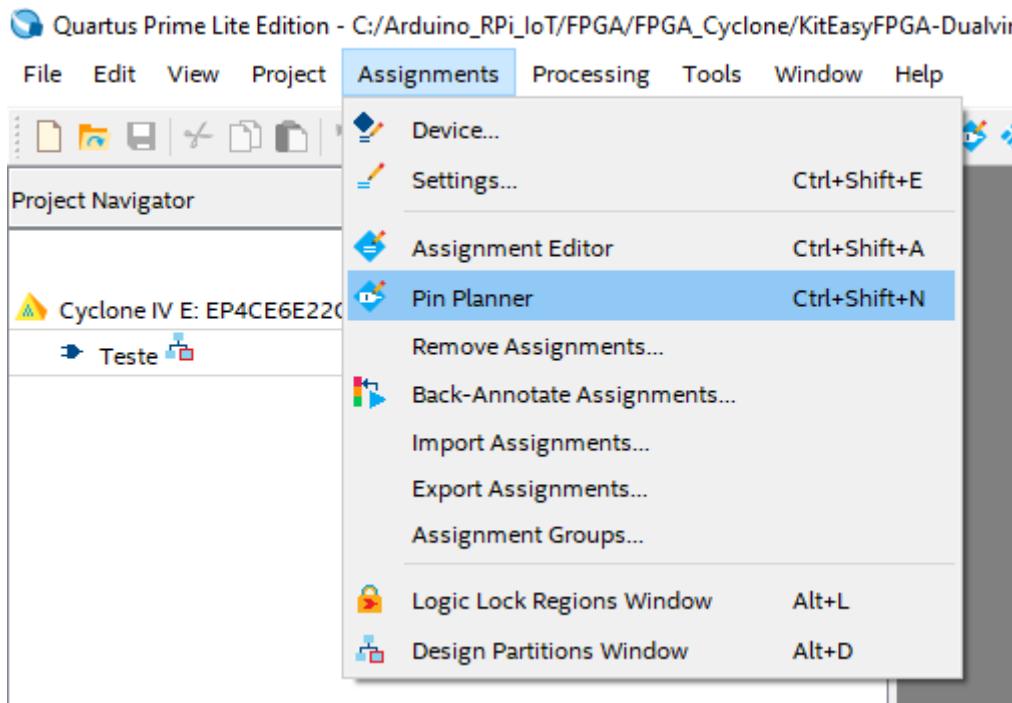


Figure 8: Assignments/Pin Planner

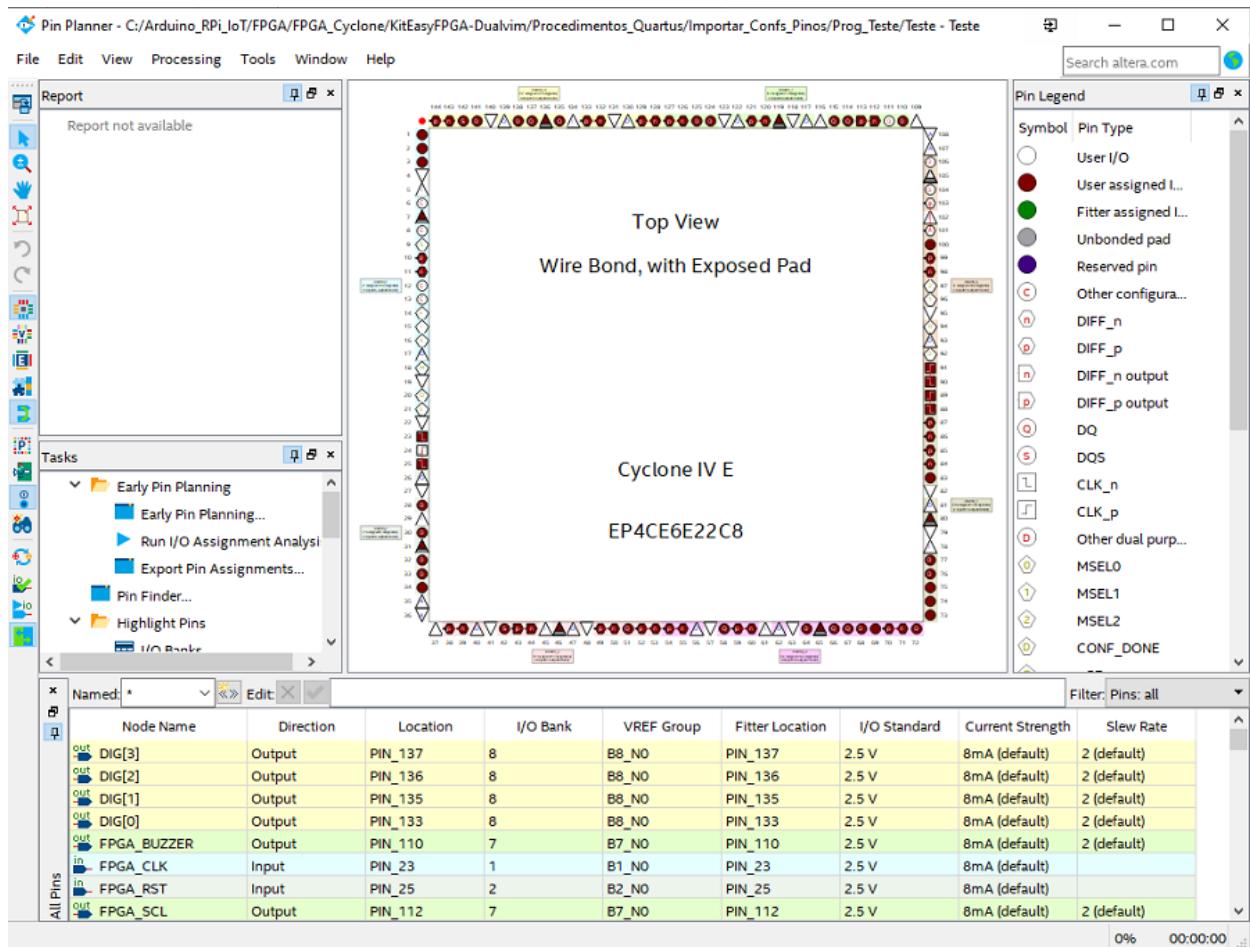


Figure 9: Tela do *Pin Planner*