

GRADO DE INGENIERÍA ELECTRONICA INDUSTRIAL Y AUTOMÁTICA

TRABAJO FIN DE GRADO

Titulo:

Propuesta de mejora de la asignatura "Informática Industrial" empleando la herramienta SIENA y simuladores 3D

Autores:

Carlos Castro Pérez Carlos Bretón Timón

Tutor:

Evelio González González

Septiembre 2016

Contenido

1	Introduc	eción	7
2	Resume	n	7
3	Abstrac	t	8
4	Metodo	logía	8
5	Ámbito	y alcance	9
6	Normat	iva	9
7	"Inform	ática Industrial"	9
8	Tercer r	nódulo y prácticas	10
9	Disposi	tivos de las prácticas de laboratorio	11
9	.1 Mie	crocontrolador ATMEGA8515	11
9	.2 Ad	aptador de comunicación ordenador-microcontrolador	12
9	.3 Dis	positivo DISEN de laboratorio	12
	9.3.1	Módulo de interruptores.	13
	9.3.2	Módulo de luces led.	14
	9.3.3	Modulo teclado numérico o teclado matricial	14
	9.3.4	Módulo display de 7 segmentos.	15
	9.3.5	Módulo del potenciómetro	16
	9.3.6	Módulo motor de corriente continua.	16
10	Estru	ctura del Proyecto	17
1	0.1 Blo	oque 1: Simulación	18
1	0.2 Blo	oque 2: Herramienta SIENA	18
11	Simu	lación del micro controlador ATMega 8515 y los módulos asociados	19
1	1.1 Mo	tivo de la simulación	19
1	1.2 Est	udio del Software especializado para el modelado 3D	19
	11.2.1	CATIA[1]:	19
	11.2.2	Inventor[2]:	20

11.	2.3	Blender[3]:	21
11.3	Est	udio del Software especializado para la simulación	22
11.	3.1	Aplicación de simulación de cada programa de modelado 3D:	22
11.	3.2	Game Maker Studio[4]:	22
11.	3.3	Unity 3D[5]:	23
11.4	Res	sultado del estudio	23
11.5	Eta	pas Simulación	24
11.6	Vei	ntajas de la simulación	25
12 E	Blend	ler	25
12.1	¿Qι	ué es Blender?	25
12.2	Ent	orno de trabajo	26
12.3	Pro	cedimiento de Blender en el proyecto	29
13 U	Jnity	3D	33
13.1	¿Qι	ué es Unity 3D?	33
13.2	Ent	orno de trabajo	33
13.3	Pro	cedimiento Unity 3D en el proyecto	34
14 A	Aplic	ación educativa para el aprendizaje de "Informática Industrial"	37
14.1	Intr	oducción	37
14.2	Obj	jetivo	37
14.3	Exp	olicación de la aplicación realizada para las prácticas de	"Informática
Indus	trial'		37
14.	3.1	Menú Principal	38
14.	3.2	Menú Microcontrolador	40
14.	3.3	Pantalla Conexión AVTISP	43
14.	3.4	Menú Módulos	44
14.	3.5	Módulo Motor	46
14	3.6	Módulo Teclado Matricial	48

Memoria

14.3.7 14.3.8		Módulo Led	50
		Módulo Interruptor	51
	14.3.9	Módulo Display	52
15	Softw	vare de aprendizaje para la enseñanza en la ingeniería	61
1	5.1 Ob	jetivos	61
1	5.2 ¿Q	ué es siena?	61
15.3 Pasos a seguir			62
	15.3.1	Registrarse en SIENA	62
	15.3.2	CompendiumLD	62
	15.3.3	Elaborar el cuestionario	63
16	Presu	puesto	69
17	7 Conclusión		69
18	8 Conclusion		70
19	9 Bibliografía		
20) Referencias		72

Anexo I Modelos 3D

Anexo II Pregunta incluidas en el sistema SIENA

Índice tablas y figuras

9-1. Placa Microcontrolador	11
9-2. Adaptador	12
9-3. Módulos	13
9-4. Módulo Interruptores	13
9-5. Módulo led	14
9-6. Esquema teclado matricial	15
9-7. Módulo display	16
9-8. Módulo potenciómetro	16
9-9. Módulo Motor	17
10-1. Estructura Proyecto	18
11-1. Interfaz Catia	20
11-2. Interfaz Inventor	21
11-3.Interfaz Blender	21
11-4. Interfaz Game Maker Studio	23
11-5. Interfaz Unity 3D	23
12-1 Menú Principal Blender	26
12-2.Menú listado	27
12-3. Herramienta Transformaciones	28
12-4. Modelos prefijados	29
12-5.Modo editar	29
12-6.Realizar un corte	30
12-7.Modificador Boolean	30
12-8. Blender cable	31
12-9. Blender Tornillo	32
12-10. Menú textura	32
13-1 Entorno Unity 3D	34
13-2. Scripts relacionados con el objeto "bitmassig"	35
13-3. Línea temporal animación	36
13-4. Esquema animaciones. Parámetros booleanos	36
14-1.Comienzo Aplicación	38
14-2.Pantalla Principal	38
14-3.Pantalla Microcontrolador	40
14-4. Pantalla Conexión AVTISP	43
14-5.Pantalla Módulos	44
14-6.Pantalla Módulo Motor	46
14-7.Pantalla Módulo Teclado Matricial	48

Carlos Bretón Timón Carlos Castro Pérez

14-8.Pantalla Módulo Led	50
14-9.Pantalla Módulo Interruptor	51
14-10.Pantalla Módulo Display	52
15-1.Mapa Conceptual	63
15-2. Crear asignatura	64
15-3. Importar mapa conceptual	64
15-4. Imagen mapa conceptual	64
15-5. Listado de nodos	65
15-6. Nueva pregunta	65
15-7. Relacionar pregunta con respuestas	66
15-8. Pregunta finalizada	67
15-9. Asociar pregunta a un nodo	67
15-10. Lista de preguntas	68
Tabla 1. Código salida	20
Tabla 2.Código cambiar escena	
Tabla 3. Código cambiar vista	
Tabla 4. Código mover con el ratón	
Tabla 5. Código enlace página web	
Tabla 6. Código texto sobre objeto	
Tabla 7.Código mover motor	
-	
Tabla 8. Código activar animaciones teclado	
Tabla 9.Código encender led	
Tabla 10.Codigo display	52
Tabla 11. Presupuesto	69

Carlos Bretón Timón Carlos Castro Pérez

Memoria

1 Introducción

Este documento es el desarrollo del trabajo de fin de grado de la carrera de Grado en Ingeniería Electrónica Industrial y Automática (GIEIA) de la Universidad de La Laguna, más relacionado con la asignatura de "Informática Industrial" de cuarto curso.

El desarrollo de este proyecto, "Propuesta de mejora de la asignatura "Informática Industrial" empleando la herramienta SIENA y simuladores 3D", se origina debido a la necesidad de que el docente, tenga una herramienta que facilite la explicación previa a la realización de las prácticas del bloque III de la asignatura "Informática Industrial". También, realizar un conjunto de cuestionarios que facilite a evaluación del alumnado en la aplicación de la Universidad de La Laguna, SIENA.

2 Resumen

Este proyecto es el documento final de la asignatura de trabajo de fin de grado (TFG). En él, se aplican todos los conocimientos obtenidos mediante la realización de la carrera (GIEIA) para ejecutar el diseño 3D de los dispositivos y microcontroladores usados en las prácticas del bloque tres de la asignatura.

Se trata de un proyecto de carácter didáctico para la asignatura de "Informática Industrial" de 4º curso del Grado de Ingeniería Electrónica Industrial y Automática. El diseño de este proyecto ha de modelarse en 3D y simular el funcionamiento del microcontrolador usado normalmente para las prácticas de la asignatura junto con el resto de dispositivos. A su vez realizar un cuestionario con la herramienta SIENA sobre el funcionamiento de éstos y de carácter general sobre la tercera parte de la asignatura, capaz de evaluar la compresión del alumno sobre el dispositivo y software de éste y el resto de conocimientos impartidos por el docente. Así mismo, convertir el tiempo de prácticas en un tiempo más eficiente a la hora de explicar éstas.

3 Abstract

The optimization of material resources are crucial for the proper operation of the laboratory practices. This project concerns the simulation of a microcontroller, in order to optimize the work in the Industrial Computing Laboratory. It is a didactic material with the purpose of helping students in the subject of "Industrial Computing" of 4th Grade Industrial Electronics and Automation Engineering. In this project we have provided all the knowledge gained during the entire stay in college.

The objective of this project is to model in 3D several components and simulate the operations of the microcontroller, normally used for the practice of the subject. Also make a concept map tool helped by SIENA, where we will show information of the work in the laboratory and finally a questionnaire. In this project, students will be presented all the necessary information and instructions for the correctly management of the microcontroller, they will be able to understand the device and the software used.

4 Metodología

La metodología a seguir durante la elaboración del proyecto será la siguiente:

- Búsqueda de información de posibles herramientas capaces de realizar una simulación en 3D.
- Análisis por módulos del dispositivo en el laboratorio, tanto dimensionalmente como en funcionamiento, así como el conjunto global de todos los componentes necesarios para realizar la práctica.
- Modelado 3D de cada una de las piezas que conforman la práctica de microcontroladores de la asignatura "Informática Industrial".
- Simular las piezas en un ambiente 3D y el funcionamiento para desarrollar una aplicación didáctica.
- Elaborar un cuestionario por medio de la herramienta SIENA.

A medida que se vaya desarrollando la simulación 3D, se irá redactando la memoria del proyecto.

Carlos Bretón Timón Carlos Castro Pérez

Memoria

5 Ámbito y alcance

Con carácter específico esta simulación tiene como objetivo facilitar la docencia del profesorado a la hora de explicar el funcionamiento en conjunto y global del microcontrolador y sus dispositivos, así como las conexiones e información de todos los componentes que intervienen en la práctica de laboratorio de la asignatura.

De esta forma se contempla, a la hora de realizar el proyecto, mostrar todas las competencias, capacidades y conocimientos adquiridos en el grado y más específicamente en la asignatura de "Informática Industrial".

En relación al alcance de la simulación, se pretende que sea una herramienta para la enseñanza. Y con respecto al desarrollo de ésta, descrito en este documento, tendrá como objetivo ser una herramienta de aprendizaje u orientación a la hora de realizar trabajos similares.

6 Normativa

Con respecto a simulaciones con carácter didáctico para microcontroladores, no hay normas específicas. Con respecto a la temática de la simulación, existe la normativa para cada componente a la hora del diseño y capacidades de cada componente.

7 "Informática Industrial"

En la asignatura "Informática Industrial" como su nombre indica, se imparten conocimientos informáticos de carácter industrial. Ésta, en la Universidad de La Laguna, en el Grado de Ingeniería Electrónica Industrial y Automática, está dividida en tres módulos impartidas por dos docentes.

El primer módulo se abarca la introducción a los conocimientos que se impartirán en la docencia de esta asignatura y el módulo segundo, abarca las comunicaciones industriales. Ambos módulos esta divididos a su vez en temas, los cuales son impartidos por el profesor Alberto Francisco Hamilton Castro el cual también es coordinador de la asignatura.

Carlos Bretón Timón Carlos Castro Pérez

Memoria

Ya el tercer módulo "Microcontroladores" impartido por Evelio José Gonzalez Gonzalez y en el cual se centra este proyecto, está dividido en un único tema que abarca una introducción y una visión general de estos dispositivos de control.

Para más información consultar:

http://eguia.ull.es/etsici/query.php?codigo=339394103

8 Tercer módulo y prácticas

El tercer módulo de la asignatura de "Informática Industrial" se denomina "Microcontroladores". En él se incluye el octavo y último tema de la asignatura. Éste abarca conocimientos relacionados con estos dispositivos de control, comenzando por una introducción a microcontroladores pasando por conocimientos más concretos como son el lenguaje ensamblador, estructura interna o el teclado matricial entre otros apartados.

Esta parte de la asignatura incluye tres prácticas en el laboratorio Lorenzo Moreno Ruiz en el edificio de la escuela de Ingeniería Informática. Están impartidas por el profesor Evelio Gonzalez Gonzalez el cual también imparte las clases teóricas de aula. Para realizar estas clases de laboratorio, el docente divide a los alumnos en grupos los cuales acudirán al laboratorio en el horario establecido y, en cada una de las prácticas, se les explicará que se realizará en ella, así como, el uso de los distintos dispositivos que intervienen en ella.

Debido a este último punto, se ha decidido convertir el tiempo de docencia de las prácticas en un tiempo más eficiente a la hora de realizar la explicación de éstas. Por ello se ha considerado realizar una aplicación 3D con carácter docente de los distintos dispositivos que intervienen en ella las cuales se detallan en este documento, así como un cuestionario en la web SIENA de la Universidad de La Laguna abarcando tanto conocimientos de las prácticas como el contenido general del tercer módulo.

Las prácticas de laboratorio constan de que, por medio del correcto conexionado computador-microcontrolador, siguiendo de manera correcta lo descrito en el guion de prácticas aportado por el docente, ejecutar de manera satisfactoria, una serie de códigos

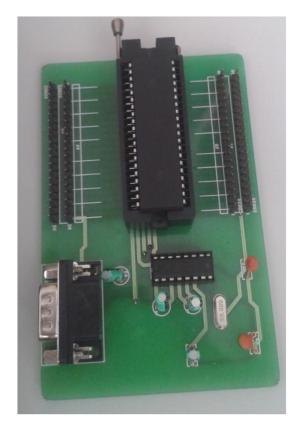
en lenguaje ensamblador que opere de manera correcta en los distintos dispositivos que intervienen en ella, siendo de mayor dificultad a medida que se realizan las tres prácticas.

Se ha visto que, a lo largo de estos años de docencia, el guion de prácticas no ha sido una herramienta capaz de transmitir su información de manera adecuada al alumnado. Por ello se ha considerado realizar una nueva herramienta con los modelos en 3D de todos los dispositivos de la práctica que facilite, de manera visual, la compresión de este documento siendo incluso el alumnado capaz de trabajar en el laboratorio en ausencia de éste.

9 Dispositivos de las prácticas de laboratorio

9.1 Microcontrolador ATMEGA8515

Éste es el microcontrolador usado en las prácticas de la asignatura de "Informática Industrial". Es un microcontrolador de ocho bits con ocho mil bytes de memoria flash programable, quinientos doce bytes de memoria EEPROM y SRAM cada uno.



9-1. Placa Microcontrolador

9.2 Adaptador de comunicación ordenador-microcontrolador

Este dispositivo es el encargado de comunicar al microcontrolador con el ordenador, haciendo de conversor del lenguaje ensamblador que trasmite para ejecutar las instrucciones descritas por el operador. En el desarrollo de la simulación se explicará con detalle la conexión adaptador /microcontrolador. La conexión ordenador/adaptador se hará por medio de un cable USB Tipo A/USB Tipo B.



9-2. Adaptador

9.3 Dispositivo DISEN de laboratorio

Siguiendo el guion de la práctica, conectaremos a los distintos módulos de este dispositivo al microcontrolador, ejecutando las instrucciones descritas en la instrucción del ordenador. Ésta será la herramienta donde ejecutaremos los códigos y conectaremos a cada módulo según lo requerido en la práctica observando los resultados.



9-3. Módulos

9.3.1 Módulo de interruptores.

El comportamiento de este es simple: dependiendo del estado del interruptor este encenderá o apagará un led. A su vez enviara una señal de su estado por el pin correspondiente. De izquierda a derecha, los ocho primeros pines corresponden de izquierda a derecha a los interruptores y los leds. Los dos pines restantes son tierra y alimentación.



9-4. Módulo Interruptores

9.3.2 Módulo de luces led.

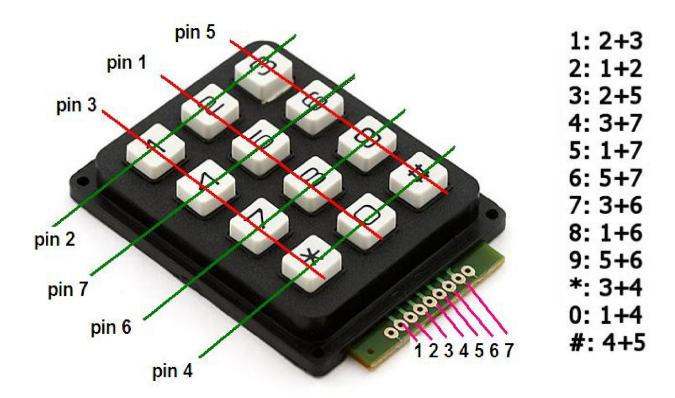
Este módulo tiene un funcionamiento similar a los interruptores. Si por el pin pasa una señal en alta, el led correspondiente se enciende. La conexión es igual que la de los interruptores sin embargo la conexión comienza desde el pin inferior.



9-5. Módulo led

9.3.3 Modulo teclado numérico o teclado matricial

Para entender las conexiones del pinado haremos uso de un esquema.

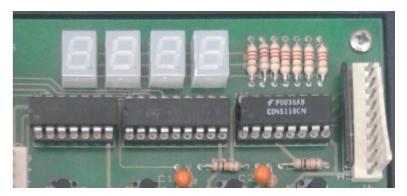


9-6. Esquema teclado matricial

El funcionamiento de cada tecla es igual a un pulsador. Para conocer el valor de la tecla pulsada hacemos uso de la tabla. Por ejemplo, si quisiera manda el valor tres, la señal pasaría por el pin dos y el cinco. De esta forma, cada tecla tiene su código distintivo, transmitido por la combinación de pines.

9.3.4 Módulo display de 7 segmentos.

De los ocho primeros pines, éstos se dividen en dos partes. Desde la parte superior, los pines del uno al cuatro son el código en binario del número a trasmitir, siendo el pin uno el menos significativo y el cuatro el más significativo. Los pines del cinco al ocho son la alimentación de cada display, donde el pin cinco es el más a la derecha y ocho el más a la izquierda.

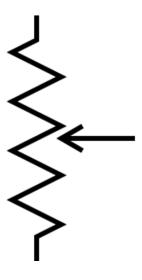


9-7. Módulo display

9.3.5 Módulo del potenciómetro

Como su nombre indica hace el funcionamiento de un potenciómetro. Si la pieza se gira a un lado o a otro el valor de la salida en el pin intermedio variará. Los pines uno y tres tiene un comportamiento de resistencia de valor fijo.



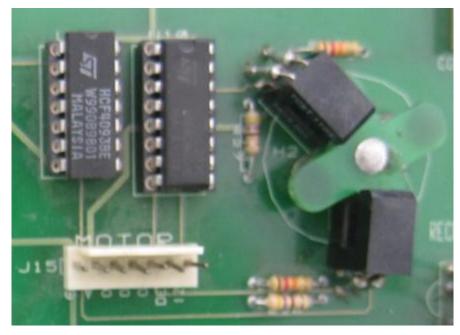


9-8. Módulo potenciómetro

9.3.6 Módulo motor de corriente continua

Dependiendo de las conexiones de los pines hace girar el motor en un sentido u otro. Los dos primeros pines son los que ejecutan el giro, anti horario el pin uno y horario el dos. Los pines cuatro y cinco son sensores que detectan si las aspas del motor han pasado por

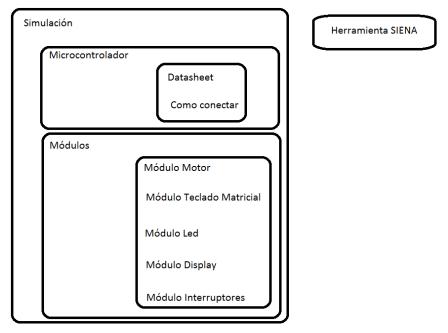
ellas siendo, el sensor inferior el pin cuatro y el superior el cinco. Los pines seis y siete son tierra y alimentación respectivamente.



9-9. Módulo Motor

10 Estructura del Proyecto

Una vez estudiado todos los puntos a tener en cuenta para realizar la mejora de la asignatura "Informática Industrial". Se decidió dividir el proyecto en dos bloques. En el primer bloque se realizará una simulación de las herramientas que se utilizan en las prácticas de la asignatura. En el segundo bloque se trabajará con la herramienta SIENA.



10-1. Estructura Proyecto

10.1 Bloque 1: Simulación

En este bloque crearemos una simulación donde el alumno podrá interactuar con el microcontrolador y los módulos asociados. La simulación la dividiremos en dos partes en la primera se explicará el microcontrolador y se detallará como conectar este a los módulos y al ordenador mediante los cables.

En la segunda parte se explicará cada uno de los módulos que intervienen en la practicas. En cada uno de estos módulos se indicará como funciona y el orden de los pines para poderlos conectar con el microcontrolador.

10.2 Bloque 2: Herramienta SIENA

En este apartado crearemos un cuestionario autodidáctico referente al tercer bloque de la asignatura "Informática Industrial" y sus prácticas. El cuestionario se realizará con la ayuda de la herramienta SIENA.

11 Simulación del micro controlador ATMega 8515 y los módulos asociados

11.1 Motivo de la simulación

La simulación consiste en crear un modelo de un sistema real para interactuar en él, ya que por motivos específicos no se pueda hacer el sistema real.

El motivo por el cual se realiza el modelado 3D y la simulación del micro controlador ATMega 8515 y los módulos asociados, es para permitir al alumno acceder a un recurso de aprendizaje de la asignatura de "Informática Industrial" del grado de ingeniería electrónica industrial y automática de la universidad de La Laguna. De esta forma el alumno estará familiarizado con las prácticas y poder realizarlas con mayor fluidez y comprensión ya que se dispone de un tiempo limitado para estas.

11.2 Estudio del Software especializado para el modelado 3D

El objetivo es encontrar un programa que realice de manera sencilla el modelado 3D. El software a elegir debe de estar en la categoría de Software CAD (ComputerAided Design) paramétrico.

Un software CAD paramétrico, es una herramienta de dibujo digital, que además de modelar en 3D y parametrizar las características de sus elementos, permite diseñar formas mucho más eficientes desde el punto de vista acústico, energético y estructural, revisando el modelo instantáneamente. De forma que, si hay una modificación en algún parámetro del modelo durante el proceso de diseño, el software calcula y optimiza automáticamente las nuevas condiciones del sistema, hasta dar con el objetivo buscado.

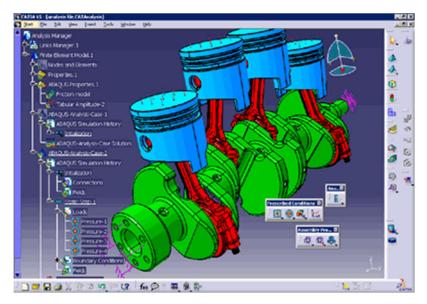
Para el estudio se analizaron los siguientes programas:

11.2.1 CATIA[1]:

Software para diseño y desarrollo de productos 3D y PLM (Product Life Management), el último término es una herramienta útil para empresas de ingeniería y diseño porque asegura una gestión eficaz de los costes del producto en todas sus etapas: creación, diseño, fabricación y lanzamiento.

Este programa está orientado al diseño avanzado de proyectos, su aplicación principal es el modelado avanzado de sólidos, superficies, ensamblaje, producción de dibujos, manufactura y análisis.

Es un programa de solución modular, su plataforma ofrece diferentes disciplinas como diseño industrial, diseño mecánico, diseño de equipos, entre otros. Una vez se escoge el diseño apropiado la interfaz de usuario permite la creación del boceto de la pieza hasta la etapa de animación. Incluye ensamblaje y planos 2D.

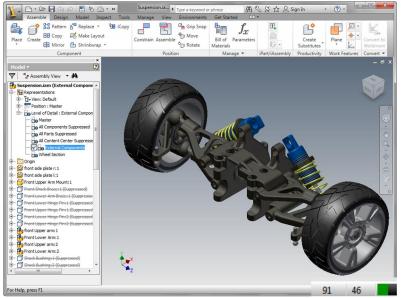


11-1. Interfaz Catia

11.2.2 Inventor[2]:

Es un software de modelado paramétrico de sólidos en 3D producido por la empresa Autodesk. Creado para el diseño mecánico de proyectos industriales. La interfaz de usuario es fácil de manejar y muy intuitiva, el entorno de desarrollo posee herramientas muy útiles tanto para el modelado 2D como para el 3D. Se logra realizar un proyecto en todas sus fases: boceto de la pieza, ensamblaje, animación, plano de componentes, materiales utilizados.

Este programa cuenta con la aplicación "Inventor Publisher" para obtener la presentación final de un proyecto, útil para situaciones donde se desea ver el despiece de las componentes.



11-2. Interfaz Inventor

11.2.3 Blender[3]:

Blender es un programa de modelado en 3D libre, apoyado por varias herramientas, es multiplataforma (corre en Windows XP, Vista 32 y 64 bits, Linux 32 y 64 bits, MacOS, Solaris, etc.). Incorpora la potente tecnología de edición de nodos que permite disponer de cada uno de los efectos por separado y no en un orden secuencial (tipo historial) donde al anular uno de ellos desaparecerían todos los posteriores.

Una vez realizado el modelo incorpora la posibilidad de dar texturas y materiales, iluminar la escena... Incluye las tecnologías más utilizadas en el diseño 3D: mallas, textos, meta-objetos, curvas, superfícies y modelado escultórico.



11-3.Interfaz Blender

11.3 Estudio del Software especializado para la simulación

En este caso se pretende buscar un programa de fácil manejo para poder realizar la simulación de los modelos hechos con uno de los programas de modelado en 3D. El resultado final ha de ser intuitivo y de fácil visualización. Además, tenemos la intención de realizar una simulación interactiva para que al usuario le sea más ameno e interesante el aprendizaje.

Para el estudio se analizaron los siguientes programas:

11.3.1 Aplicación de simulación de cada programa de modelado 3D:

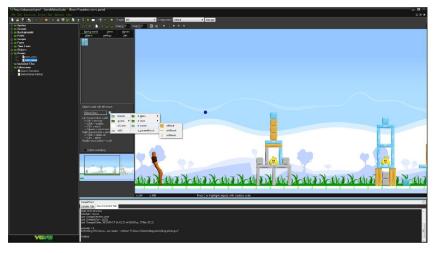
Los tres programas de modelado capaces de simular sus propios modelos. Cada uno posee una distinta forma de simulación siendo el programa Blender el más detallado.

En el caso del Inventor y del Catia la simulación se logra por bloques. E l movimiento del modelo se logra variando las restricciones que limitan los grados de libertad de las piezas, en un tiempo específico. Básicamente la trayectoria de movimiento se obtiene definiendo tanto la posición inicial como la posición final de la componente.

En el caso del Blender además de poder realizar la simulación igual que el Catia y el Inventor. También tienes la posibilidad de realizar la simulación dando a los modelos propiedades físicas y de esta forma la simulación en más fluida.

11.3.2 Game Maker Studio[4]:

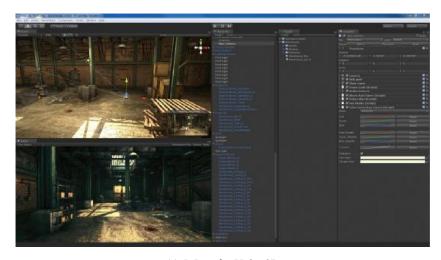
Programa para realizar videojuegos. Permite crear casi cualquier tipo de juego (incluso 3D) y existe muchísima documentación en Internet. Su lenguaje de scripting es opcional y hace que gane mucha profundidad.



11-4. Interfaz Game Maker Studio

11.3.3 Unity 3D[5]:

Al igual que el Game Maker Studio es un programa para realizar videojuegos. Permite crear todo tipo de juegos, aunque está pensada para el desarrollo 3D y es muy versátil. Este programa es mucho más conocido que el Game Maker Studio.



11-5. Interfaz Unity 3D

11.4 Resultado del estudio

Los programas analizados en este proyecto, presentan características de diseño muy similares en cuanto a las herramientas para la creación de modelos en 3D, por lo que respecto a este tema todos son muy eficaces e igual de útiles. Pero nos dimos cuenta que tanto el Catia y el Inventor son programas para realizar modelos de piezas mecánicas, y no como en nuestro caso que tenemos que realizar modelos electrónicos.

Otra cuestión motivo de análisis, fue el entorno de simulación. Como idea principal queríamos hacer una simulación interactiva, por eso nos decantamos por el software para realizar videojuegos.

Finalmente, el software elegido para realizar la simulación es el Unity 3D. Por comodidad, la interfaz de usuario es muy fácil de manejar y de fácil aprendizaje. Por otro lado, optamos por el Blender como software de modelado ya que cumple con las características de ser un software de modelado paramétrico y realiza perfectamente modelos en 3D.

Otro punto importante es que ambos softwares son compatibles. Los modelos realizados por Blender, el Unity no tienen ningún problema para leerlos. Otro tema a tener en cuenta fue facilidad para adquirir los programas ya que ambos son libres y de esta forma no tener problemas con las licencias.

11.5 Etapas Simulación

Se llegó a la conclusión que las etapas más relevantes para el proyecto fueron:

Definición del sistema

El sistema real simulado fue el Microcontrolador ATMega8515 y sus módulos.

Variables a concluir en el modelo

Indagamos, preguntando a los alumnos, respecto a las dificultades que tenían en el momento de realizar las practicas. Las conclusiones fueron que tenían dificultad con la teoría y también en el momento de conectar los módulos entre sí, cableado de pines.

Recolección de datos

Se efectúo la medida de cada pieza que componía el conjunto y se tomaron fotos de su aspecto para lograr un modelo lo más semejante a sistema real.

Carlos Bretón Timón Carlos Castro Pérez

Memoria

Programación del modelo

Realizamos una simulación donde el alumno tuviera la teoría más accesible, además de indicar de una forma detallada como se ha de comunicar los distintos módulos del micro controlador.

Verificación del modelo

Obtenido modelo se verificó la similitud con el sistema real.

11.6 Ventajas de la simulación

Usando la simulación como apoyo a la enseñanza se logra un método de estudio más interactivo y didáctico. El alumno ganará agilidad en la realización de las practicas, logrando reducir el tiempo en realizar las practicas. Además, este método es más entretenido, de esta forma el alumno se divertirá, aumentando el tiempo que este estará enfocado con la asignatura.

Las grandes empresas actualmente utilizan el diseño y definición de un modelo en 3D, para analizar y evaluar un producto antes de ser fabricado, generando una reducción de tiempo y costo.

Otro punto importante es que el alumno no tiene que estar en el laboratorio para poder trabajar con el modelo simulado, disminuyendo en este caso la carga de horarios que se le somete al espacio. El trabajar en el diseño de un modelo virtual, no implica interrumpir las operaciones.

12 Blender

12.1 ¿Qué es Blender?

Blender es un programa dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. En Blender se puede desarrollar vídeo juegos ya que posee un motor de juegos interno.

El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente aunque posteriormente pasó a ser software libre. Las capacidades de Blender son:

modelado 3D, mapeado UV, texturización, edición de gráficos, rigging and skinning, simulación de fluidos y partículas, simulación cuerpo blando, animación, seguimiento de cámara, renderizado, edición y composición de video.

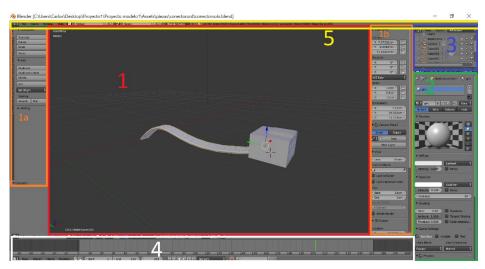
Por medio del siguiente enlace se puede acceder a más información del programa: https://en.wikipedia.org/wiki/Blender (software)

12.2 Entorno de trabajo

Para poder explicar los procesos realizados para hacer cada uno de los modelos creados en Blender mostramos una pequeña introducción a la interfaz del programa.

El software Blender dispones de dos entornos. El primero está dedicado al modelado y el segundo a la simulación. Solo explicaremos el entorno de modelado, ya que es el que usamos para nuestro proyecto. En el caso de la simulación explicaremos más adelante el entorno de Unity 3D, programa con el que la realizamos.

Mostraremos cada una de las diferentes áreas de trabajo en las que se divide el entorno de modelado del Blender.



12-1 Menú Principal Blender

Referente a la imagen 12-1, el área central (1) es el espacio destinado a la visualización del proyecto 3D. El nombre concreto del editor es Vista 3D. En esta área se pueden hacer visibles o no visibles dos cuadros con gran variedad de recursos.

El cuadro dentro de Vista 3D situado a la derecha (1a) se denomina Herramientas. Se puede activar o desactivar. El cuadro que aparece en la parte derecha de Vista 3D (1b), y que no aparece en Blender por defecto, se denomina Propiedades Básicas Modelo.

En la parte derecha de la interfaz está uno de los editores más usados; se trata del Panel (2). En esta zona nos encontraremos unos paneles capaces de editar el modelo.

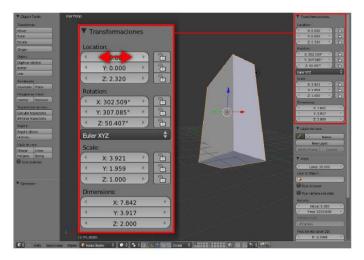
Encima de los paneles se encuentra el menú Listado (3). Permite acceder a cada elemento de la escena con rapidez además de determinar desde allí algunas de sus propiedades como la visibilidad. En la imagen 12-2 se nuestra el menú listado por defecto, hay una cámara (Camera), un cubo (Cube) y una lámpara (Lamp) además de un entorno (World).



12-2.Menú listado

La zona inferior de Vista 3D está destinada al editor Línea de tiempo (4). Recordemos que Blender es una herramienta que permite animación. Desde aquí se concreta en qué fotograma nos encontramos tan solo con desplazar la línea vertical de color verde. Y en la zona alta de la interfaz se encuentra el editor Info (5) donde se puede encontrar el clásico menú Archivo.

Como en cualquier programa de modelado los sólidos tienen propiedades. Podemos acceder a estas propiedades en la barra de herramientas "Transformaciones". Aquí queda constancia del valor de cada transformación geométrica elemental (escala, movimiento, giro) así como de cada variación de tamaño en el objeto. Para editar uno de estos campos con flechas a los lados, podemos acceder con clic de ratón o mover horizontalmente como si fuera una barra deslizadora.



12-3. Herramienta Transformaciones

Los controles básicos de transformación del modelo dentro de la vista 3D son:

- Mover. Para desplazar un objeto basta con usar el atajo de teclado "G".
- Rotar y Escalar. Para poder rotar y escalar los atajos son "R" y "S" respectivamente.
- Borrar. Para optimizar el trabajo usaremos la tecla "Supr" o la tecla "X".
- <u>Duplicar</u>. La tecla rápida es "Shift_D".

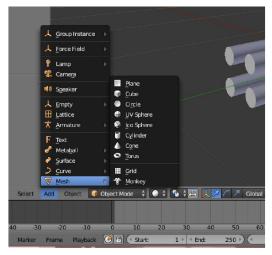
El editor Paneles es uno de los lugares con mayor cúmulo de opciones editables. Pasamos a describir solo los referentes a nuestro proyecto:

- Render. El término render está completamente arraigado en la jerga del diseño 3D y hace referencia al proceso por el cual el modelado 3D se convierte en una imagen fija, en una animación, en una simulación... También llamamos *render* al propio resultado.
- <u>Mundo.</u> Aquí se determinan aspectos que van a tener que ver con el ambiente que envuelve la escena. Un buen ejemplo de lo que se edita aquí son las estrellas del firmamento, la niebla o algunas iluminaciones de carácter global no basadas en puntos luminosos.
- <u>Objeto</u>. En este panel se editan asuntos como la relación entre dos objetos, por ejemplo, una relación de parentesco por la que un objeto queda ligado a otro.
- <u>Modificadores</u>. Estas son importantísimas herramientas destinadas a ahorrar buenas dosis de trabajo. Desde aquí se crea de manera automática una animación de un objeto que late como un corazón, sólo por poner un ejemplo.

- <u>Datos de objeto</u>. Dentro de un objeto es posible distinguir entre una zona u otra a pesar de formar parte de la misma malla.
- <u>Material</u>. Para que un objeto parezca más o menos real habrá que indagar en sus propiedades: color, reflexión...
- <u>Texturas</u>. No siempre es suficiente con un material bien diseñado para ganar realismo. Las texturas aportan nuevas cualidades al material.

12.3 Procedimiento de Blender en el proyecto

Con el programa Blender se hicieron un total de 38 modelos para simular tanto el microprocesador como los módulos relacionados con este.

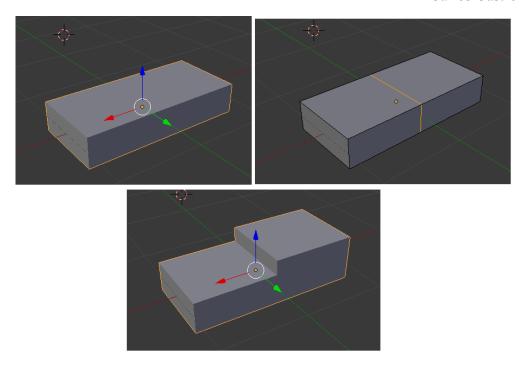


12-4. Modelos prefijados

Para comenzar a trabajar se parte de un objeto ya prefijado (cubo, esfera, cilindro, ...) (imagen 12-4). Una vez seleccionado el objeto se tiene la posibilidad de modificar su posición en la pantalla, la rotación, las dimensiones y su escala. Después de modificar estos parámetros pasamos del "modo objeto" al "modo editar". En este modo se puede seleccionar y modificar los vértices, las líneas y los planos de la pieza gracias a los comandos descritos en el apartado anterior (mover, rotar y escalar). Además, se puede realizar un corte al objeto si es necesario.

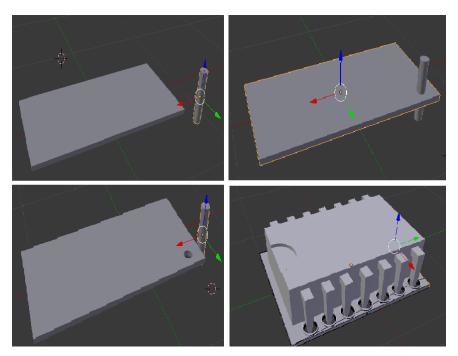


12-5.Modo editar



12-6.Realizar un corte

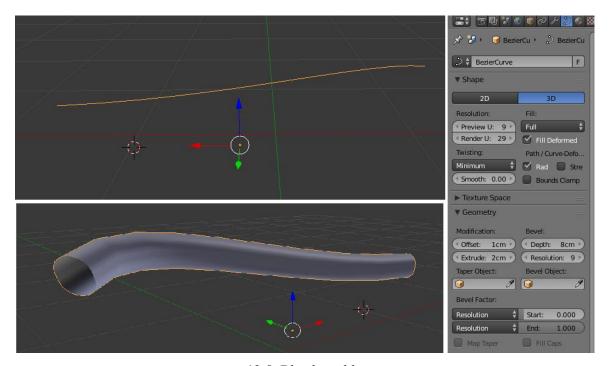
También el objeto se puede modificar con el panel de "modificadores". Unos de los más usados en este proyecto es el "Boolean", este modificador te permite añadir o quitar un objeto a otro objeto. (imagen 12-6).



12-7.Modificador Boolean

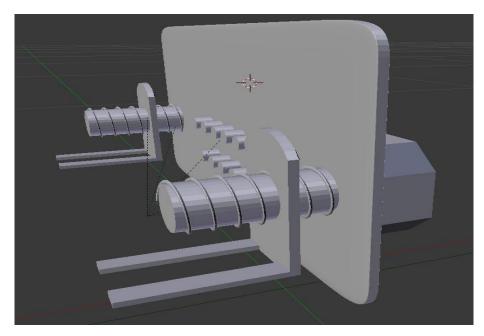
Realizar los modelos con Blender es un trabajo sencillo, una vez que te has familiarizado con el programa, pero largo. Blender por medio del menú listado, tienes la posibilidad de activar y desactivar las piezas del modelo (imagen 12-2). También puedes agrupar las piezas de los modelos que ayuda bastante en la organización del proyecto.

Uno de los inconvenientes que aparecieron fue en el momento de crear los cables. En un primer intento se intentó solucionar con los cilindros, pero daba la sensación de que los cables eran demasiado rígidos y no aparentaban ser cables. Después de indagar lo descubrimos otra manera de realizarlos gracias a la herramienta "curva Beizer". Esta herramienta nos permite crear una curva con todos los puntos necesarios para luego poder cambiar el ancho.



12-8. Blender cable

Otro inconveniente fue cuando realizamos el tornillo de la placa del micro controlador. Los solucionamos con el modificador "Screw".



12-9. Blender Tornillo

Por último, a cada una de las piezas del modelo se le añadió una textura para posteriormente en Unity 3D poderlas colorear del color adecuado.



12-10. Menú textura

Carlos Bretón Timón Carlos Castro Pérez

Memoria

13 Unity 3D

13.1 ¿Qué es Unity 3D?

Unity 3D es una herramienta que nos ayuda a desarrollar videojuegos para diversas plataformas mediante un editor y scripting para crear videojuegos con un acabado profesional. Esta herramienta está accesible al público en diferentes versiones, gratuita y profesional.

Estas versiones están enfocadas para desarrollar videojuegos para PC, Mac y Web. Sin embargo, además de estas versiones "básicas" existen añadidos que permiten trasladar el desarrollo a dispositivos móviles.

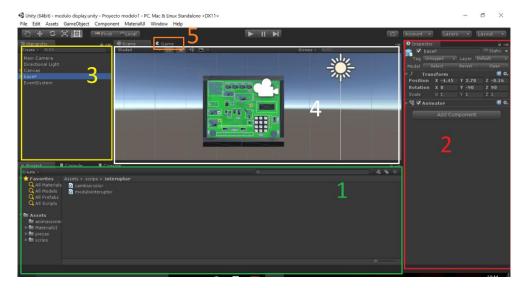
Unity 3D es un editor visual muy útil y completo donde podremos importar nuestros modelos 3D, texturas, sonidos, etc. para después ir trabajando con ellos. Además, incluye la herramienta de desarrollo MonoDevelop con la que podremos crear scripts en JavaScript y C#. También podemos encontrar tutoriales y recursos en su web oficial. Si se desea conocer más sobre Unity 3D pueden acceder al siguiente enlace:

https://es.wikipedia.org/wiki/Unity (software)

13.2 Entorno de trabajo

Inicialmente Unity era un entorno de desarrollo de juegos 3D. Se podían desarrollar entornos y juegos 2D ajustando los parámetros de tu juego para simular 2D.

El editor de Unity 3D es uno de los más sencillos y potentes del mercado. Se divide en 5 vistas principales:



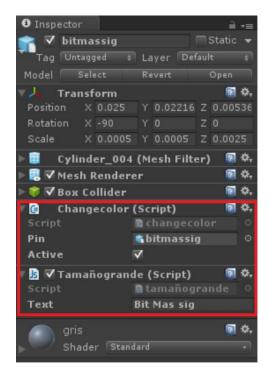
13-1 Entorno Unity 3D

- Explorador (1): Lista todos los elementos (o activos) de tus proyectos. Permite
 ordenar de forma sencilla la aplicación. En esta vista se encuentran las imágenes,
 escenas, scripts, audios, prefabs, texturas, atlas y todos los elementos que usarás o
 podrás usar en tu juego o aplicación.
- Inspector (2): Muestra y define las propiedades de los elementos del proyecto. Modifica valores de forma rápida, cambia texturas arrastrando ficheros desde el Explorador, añade scripts, guarda prefabs, ...
- Jerarquía (3): Lista jerárquica de los elementos de la escena.
- Escena (4): Diseño y maqueta del juego completo o una pantalla o sección de éste. Cada escena representa un nivel o sección diferente del juego (portada, nivel 1, nivel 2, login, ...).
- Juego (5): Visualiza tu juego a distintas resoluciones.

13.3 Procedimiento Unity 3D en el proyecto

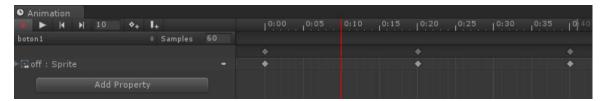
Una vez explicado el entorno de Unity 3D, mostraremos los pasos que se siguieron para realizar la aplicación. En primer lugar, se creó el menú, se siguió la organización comentada en el apartado seis bloques uno (se puede ver el esquema en la imagen 10-1). El programa tiene una herramienta especializada en crear menú llamada "User interface". con esta herramienta podemos crear botones y paneles creando de esta manera un menú. Además, esta herramienta tiene un lenguaje propio para poner movernos entre escenas o activar los botones.

Después de crear el menú colocamos cada uno de los modelos en su escena correspondiente. Los realizamos de la forma comentada anteriormente (imagen 10-1), cada módulo y cada parte del microcontrolador se colocó en una escena distinta y a través del menú se es capaz de navegar entre ellas. Unity 3D te permite crear las escenas solo usando los scripts, pero se decidió crear cada una de las escenas colocando los modelos en ellas y solo usando los scripts para programar las interacciones con el usuario. El programa está destinado a lenguaje de objetos por lo que simplemente solo hace falta relacionar el objeto con el script deseado.

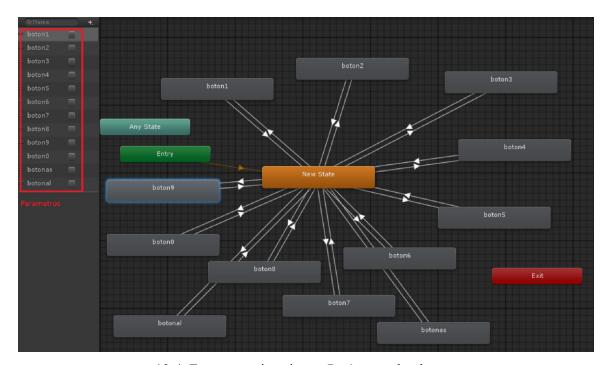


13-2. Scripts relacionados con el objeto "bitmassig"

En una de las escenas se creó una animación. Para hacer una animación primero se crea una línea temporal con cada una de las animaciones que se desea. Luego se realiza un esquema con las animaciones y se le asigna a cada una de las transiciones de las animaciones un parámetro de tipo booleano. Por último, por medio de un script se activa la animación que se desee.



13-3. Línea temporal animación



13-4. Esquema animaciones. Parámetros booleanos

14 Aplicación educativa para el aprendizaje de "Informática Industrial"

14.1 Introducción

El uso de softwares multimedia enfocados a la enseñanza cada día está más difundido. Este método es usado para aumentar la motivación de los alumnos y mejorar la calidad de la educación.

De este modo se puede decir que un software educativo es una herramienta empleada para el aprendizaje, con el fin de mejorar los conceptos principios y habilidades adquiridas durante el proceso de enseñanza. Por lo que conceptos complejos, ayudados de este medio, pueden ser más fáciles de asimilar.

Los alumnos utilizan programas especializados para la materia a tratar, donde disponen de la información necesaria y actividades.

14.2 Objetivo

Con la aplicación realizada con el software Unity 3D se intentará, en lo máximo posible, ayudar a los alumnos matriculados en la asignatura de "Informática Industrial" del Grado de Ingeniería Electrónica Industrial y Automática. Desafortunadamente en un cuatrimestre no es posible adquirir todos los conocimientos posibles sobre una materia tan amplia como es esta.

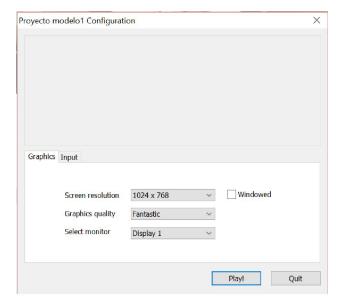
Con esta aplicación se pretende mejorar los conceptos adquiridos y agilizar las prácticas de la asignatura. El alumno podrá tener un primer contacto con el micro controlador usada en la practicas y los módulos asociados a esta.

Con este objetico en mente, se podrá acceder en la aplicación a conceptos conflictivos que les surgen a los alumnos durante las prácticas, donde se explica de forma detallada como proceder.

14.3 Explicación de la aplicación realizada para las prácticas de "Informática Industrial"

El alumno podrá acceder a la aplicación por medio de un archivo .EXE creado por el software Unity 3D.

Una vez ejecutado el archivo le aparecerá una pantalla donde simplemente con seleccionar "play" entrarán en la aplicación.



14-1. Comienzo Aplicación

Una vez realizado estos primeros pasos nos aparecerá el menú principal de la aplicación. En esta pantalla podemos ver en modelado 3D todos objetos que se van a usar en la practicas (el microcontrolador, los módulos y los cables).

14.3.1 Menú Principal



14-2.Pantalla Principal

El alumno podrá interactuar con el modelo con el ratón, de esta forma la experiencia será más intuitiva y fácil. También se podrá mover a través de la aplicación por medio de un menú sencillo compuesto por botones. Por último, en cada momento puede acceder a una guía explicativa, por medio de un botón de "Instrucciones", donde se le informará como proceder en cada pantalla.

Tabla 1. Código salida.

```
public class Exit : MonoBehaviour
{
    // Salir aplicación
    public void Exits()
    {
        Application.Quit();
    }
}
```

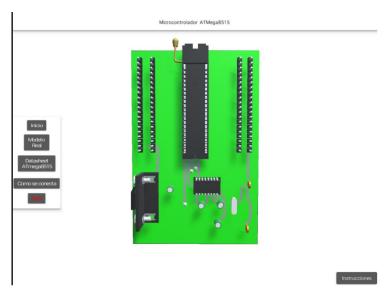
Tabla 2.Código cambiar escena

```
public class LoadOnClick : MonoBehaviour
{
    // Carga escena = level
    public void LoadScene(int level)
    {
        SceneManager.LoadScene(level);
    }
}
```

Este es el código usado para el menú de botones. En el primer caso sirve para salir de la aplicación si pulsamos el botón asignado. Gracias al código siguiente podemos navegar por las diferentes pantallas de la aplicación. Cada pantalla tiene como referencia un número y dependiendo del número que le asignemos a "level" iremos a la pantalla designada.

Se ha divido la aplicación en dos partes una asignada a la explicación del micro controlador y la otra a cada uno de los módulos que se trabaja en la práctica. Desde la pantalla principal se puede acceder a estas dos partes.

14.3.2 Menú Microcontrolador



14-3. Pantalla Microcontrolador

En esta pantalla el alumno podrá mover el modelo 3D en una vista tridimensional y poder acercar la vista, para poder ver en detalle cada componente. Además, puede acceder a una fotografía del modelo real para comparar con el modelo 3D.

Tabla 3. Código cambiar vista

```
public class DragMouseOrbit : MonoBehaviour
{

// Variables

public Transform target;

public float distance = 5.0f;

public float xSpeed = 120.0f;

public float ySpeed = 120.0f;

public float yMinLimit = -20f;

public float yMaxLimit = 80f;

public float distanceMin = .5f;

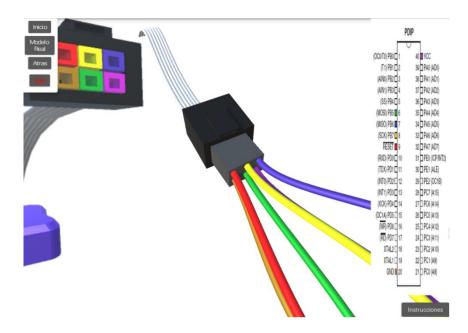
public float distanceMax = 15f;
```

```
public float smoothTime = 2f;
  float rotation YAxis = 0.0f;
  float rotationXAxis = 0.0f;
  float velocityX = 0.0f;
  float velocityY = 0.0f;
  // Inicialización
  void Start()
    Vector3 angles = transform.eulerAngles;
    rotationYAxis = angles.y;
    rotationXAxis = angles.x;
    // Permite que el cuerpo rígido no cambien en la rotación
    if (GetComponent<Rigidbody>())
       GetComponent<Rigidbody>().freezeRotation = true;
    }
  // Está pendiente en todo momento
  void LateUpdate()
    if (target)
       if (Input.GetMouseButton(0))
         velocityX += xSpeed * Input.GetAxis("Mouse X") * distance * 0.02f;
         velocityY += ySpeed * Input.GetAxis("Mouse Y") * 0.02f;
       rotationYAxis += velocityX;
       rotationXAxis -= velocityY;
       rotationXAxis = ClampAngle(rotationXAxis, yMinLimit, yMaxLimit);
       Ouaternion
                     fromRotation
                                          Quaternion.Euler(transform.rotation.eulerAngles.x,
transform.rotation.eulerAngles.y, 0);
       Quaternion toRotation = Quaternion.Euler(rotationXAxis, rotationYAxis, 0);
       Quaternion rotation = toRotation;
       distance = Mathf.Clamp(distance - Input.GetAxis("Mouse ScrollWheel") *
```

```
distanceMin, distanceMax);
       RaycastHit hit;
       if (Physics.Linecast(target.position, transform.position, out hit))
         distance -= hit.distance;
       Vector3 negDistance = new Vector3(0.0f, 0.0f, -distance);
       Vector3 position = rotation * negDistance + target.position;
       transform.rotation = rotation;
       transform.position = position;
       velocityX = Mathf.Lerp(velocityX, 0, Time.deltaTime * smoothTime);
       velocityY = Mathf.Lerp(velocityY, 0, Time.deltaTime * smoothTime);
     }
  }
  // Transforma el ángulo
  public static float ClampAngle(float angle, float min, float max)
  {
    if (angle < -360F)
       angle += 360F;
    if (angle > 360F)
       angle = 360F;
    return Mathf.Clamp(angle, min, max);
  }
```

Gracias a este código somos capaces de cambiar la vista de la cámara respecto a un objeto. En primer lugar, gracias a la clase "Start ()", es una clase predeterminada por el software Unity3D que se ejecuta siempre al comienzo de la aplicación, asignamos el objeto por medio del comando "GetComponent". Una vez asignado el objeto que queremos observar con el comando "Input.GetMouseButton(0)" el programa está atento a los eventos que realiza el ratón y gracias a los comandos "Quaternion" y "vector3" transforma los datos a las componentes X, Y, Z.

14.3.3 Pantalla Conexión AVTISP



14-4. Pantalla Conexión AVTISP

Se puede acceder a esta pantalla desde el menú del microcontrolador. Se comprobó que el alumno tenía problemas en el momento de conectar el AVTISP mkII con el microcontrolador. En esta pantalla además de tener un gráfico de la conexión el alumno puede mover la vista de cámara de la misma forma que la pantalla del "microcontrolador".

Ahora se explicará el apartado de los módulos que podemos acceder desde la pantalla principal.

14.3.4 Menú Módulos



14-5.Pantalla Módulos

Desde esta pantalla por medio del menú se puede acceder a las explicaciones de cada módulo que se realizan en las practicas. En esta pantalla el usuario puede mover el modelo haciendo "drag" con el ratón. Además, podrá interactuar con cada componente haciendo click derecho, al realizar esta acción el alumno podrá acceder al datasheet del componente seleccionado. Por último, se podrá comparar el modelo 3D con una fotografía del modelo real.

Tabla 4. Código mover con el ratón

```
public class movewithmouse : MonoBehaviour

{

// Variables

private Vector3 screenPoint;

private Vector3 offset;

// Clase propia del Unity (llama al ratón)

void OnMouseDown()
```

```
{
    screenPoint = Camera.main.WorldToScreenPoint(gameObject.transform.position);

    offset = gameObject.transform.position - Camera.main.ScreenToWorldPoint(new Vector3(Input.mousePosition.x, Input.mousePosition.y, screenPoint.z));
}

// Clase propia del Unity (Ilama al ratón)
    void OnMouseDrag()
{
        Vector3 curScreenPoint = new Vector3(Input.mousePosition.x, Input.mousePosition.y, screenPoint.z);
        Vector3 curPosition = Camera.main.ScreenToWorldPoint(curScreenPoint) + offset;
        transform.position = curPosition;
}
```

"OnMouseDown" y "OnMouseDrag" son clases predeterminadas por el programa.
"OnMouseDown" guardamos las coordenadas actuales del objeto. Mientras que
"OnMouseDrag" movemos el objeto según el movimiento del puntero del ratón.

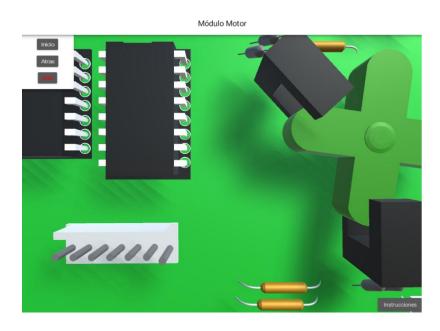
Tabla 5. Código enlace página web

```
public class onclickgoweb : MonoBehaviour {
   public string webe;

// Clase propia del Unity (llama al ratón)
   void OnMouseOver() {
      if (Input.GetMouseButtonUp(1))
      {
            //Abrir web
            Application.OpenURL(webe);
      }
      }
   }
}
```

Con estas instrucciones accedemos al datasheep de cada componente. El programa está atento a los eventos del ratón y cuando pulsamos con el boton izq. nos envía a la dirección web almacenada en la variable string "webe".

14.3.5 Módulo Motor



14-6.Pantalla Módulo Motor

En esta pantalla explica los pines del módulo. En cada pin, al colocarse con el ratón encima, aparece un texto indicando lo que hace cada uno. Además, al colocarse sobre el pin de alimentación del motor, el usuario podrá observar como las aspas girar tanto hacia la izquierda como a la derecha, dependiendo del pin seleccionado.

Tabla 6. Código texto sobre objeto

```
function Start()
{
// Variables
    guiStyleFore = new GUIStyle();
    guiStyleFore.normal.textColor = Color.red;
    guiStyleFore.alignment = TextAnchor.UpperCenter;
    guiStyleFore.wordWrap = true;
    guiStyleBack = new GUIStyle();
```

```
guiStyleBack.normal.textColor = Color.red;
  guiStyleBack.alignment = TextAnchor.UpperCenter;
  guiStyleBack.wordWrap = true;
// Clase propia del Unity (llama al ratón)
function OnMouseEnter ()
{
  currentToolTipText = text;
// Clase propia del Unity (llama al ratón)
function OnMouseExit ()
  currentToolTipText = "";
}
// Clase propia del Unity (User Interface)
function OnGUI()
  if (currentToolTipText != "")
  {
    var x = Event.current.mousePosition.x;
    var y = Event.current.mousePosition.y;
    GUI.Label (Rect (x-149,y+40,300,60), currentToolTipText, guiStyleBack);
    GUI.Label (Rect (x-150,y+40,300,60), currentToolTipText, guiStyleFore);
  }
```

Estas instrucciones sirven para que aparezca texto es el objeto asignado cuando el ratón esta encima de este.

Tabla 7. Código mover motor

```
public class moverdere : MonoBehaviour
{
    public GameObject ventilador;
```

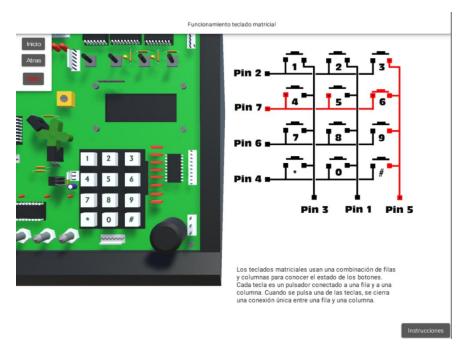
```
// Clase propia del Unity (llama al ratón)

void OnMouseOver()
{

ventilador.transform.Rotate(new Vector3(0, 0, 10));
}
```

Cuando el ratón está encima del pin asignado el "ventilador" gira. Para cambiar la dirección de giro solo hay que cambiar las coordenadas del "Vector3".

14.3.6 Módulo Teclado Matricial



14-7. Pantalla Módulo Teclado Matricial

En esta pantalla el alumno podrá informarse de cómo funciona un teclado matricial. Al seleccionar cualquier número del teclado se activará una animación indicando la conexión de ese determinado número.

Tabla 8. Código activar animaciones teclado

```
public class click : MonoBehaviour {

// Variables para animación

public string animaact;
```

```
public string animadesact1;
public string animadesact2;
public string animadesact3;
public string animadesact4;
public string animadesact5;
public string animadesact6;
public string animadesact7;
public string animadesact8;
public string animadesact9;
public string animadesact10;
public string animadesact11;
public GameObject ani;
public Animator animator;
// Inicialización
void Start()
{
  animator = ani.GetComponent<Animator>();
}
// Clase propia del Unity (llama al ratón)
void OnMouseOver(){
  bool uno = animator.GetBool(animaact);
  if (Input.GetMouseButtonDown(0) && uno == false)
    {
    animator.SetBool(animaact, true);
    animator.SetBool(animadesact1, false);
    animator.SetBool(animadesact2, false);
    animator.SetBool(animadesact3, false);
    animator.SetBool(animadesact4, false);
    animator.SetBool(animadesact5, false);
    animator.SetBool(animadesact6, false);
    animator.SetBool(animadesact7, false);
    animator.SetBool(animadesact8, false);
    animator.SetBool(animadesact9, false);
    animator.SetBool(animadesact10, false);
```

```
animator.SetBool(animadesact11, false);
}
if (Input.GetMouseButtonDown(0) && uno == true)
{
    animator.SetBool(animaact, false);
}
}
```

Al seleccionar el botón de cada número se activará la animación correspondiente y se desactivará las demás.

14.3.7 Módulo Led



14-8. Pantalla Módulo Led.

En esta pantalla se indica cada pin y se encenderá el led asignado a ese pin.

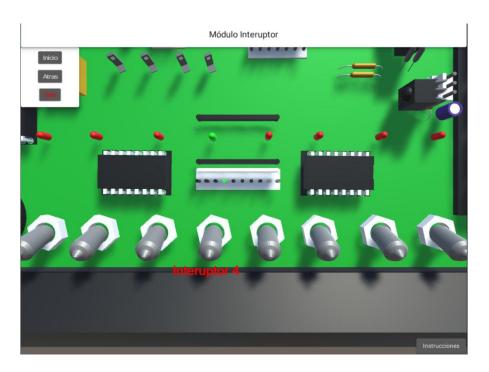
Tabla 9.Código encender led

```
public class changecolorled : MonoBehaviour
{
```

```
// Variables
public GameObject led;
// Clase propia del Unity (Ilama al ratón)
void OnMouseEnter()
{
    led.GetComponent<Renderer>().material.color = Color.green;
}
// Clase propia del Unity (Ilama al ratón)
void OnMouseExit()
{
    led.GetComponent<Renderer>().material.color = Color.red;
}
```

Cuando el ratón entra en el pin el led cambia al color verde. Por el contrario, cuando el ratón sale vuelve al color original.

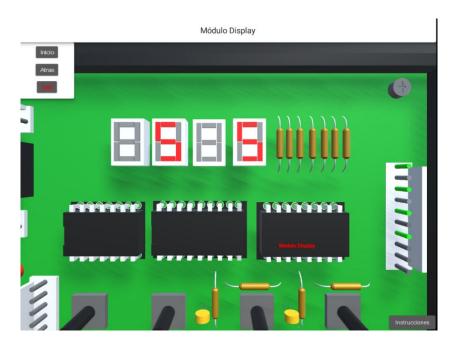
14.3.8 Módulo Interruptor



14-9.Pantalla Módulo Interruptor

En este módulo se ha realizado lo mismo que en el módulo led. Cuando el ratón se sitúa encima del interruptor se indica el led y el pin correspondiente a este. También se ha usado el mismo código para cambiar el color de los leds y del pin.

14.3.9 Módulo Display



14-10.Pantalla Módulo Display

En esta parte se pretende explicar cómo funciona este módulo. Para ello el usuario puede activar cada pin, con el botón derecho del ratón, y de esta forma activar y desactivar el display asignado a cada pin. También puede activar los pines correspondientes a los bits del display, siendo el primero el menos significativo y el cuarto el más significativo. Dependiendo de los bits activados cambiará el número de los displays.

Tabla 10.Codigo display

```
public class displaynumber : MonoBehaviour {
    // Variables
    public GameObject pinact;
    public GameObject pinmenossig;
    public GameObject pin2menos;
    public GameObject pin3menos;
```

```
public GameObject pinmassig;
  public GameObject disbot;
  public GameObject disbotleft;
  public GameObject disbotrig;
  public GameObject dismid;
  public GameObject distop;
  public GameObject distopleft;
  public GameObject distoprig;
  // Activar numero del display según botón
  public void numero()
  {
    //0
    if
        (pinmenossig
                        .GetComponent<Renderer>().material.color
                                                                          Color.grey
pin2menos.GetComponent<Renderer>().material.color
                                                                      Color.grey
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                      Color.grey
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
    {
      disbot .GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.grey;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.red;
      distop.GetComponent<Renderer>().material.color = Color.red;
    }else
    //1
         (pinmenossig.GetComponent<Renderer>().material.color
                                                                         Color.green
                                                                                       &
pin2menos.GetComponent<Renderer>().material.color
                                                            ==
                                                                      Color.grey
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                      Color.grey
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
      disbot.GetComponent<Renderer>().material.color = Color.grey;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.grey;
```

```
dismid.GetComponent<Renderer>().material.color = Color.grey;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }else
    //2
    if
         (pinmenossig.GetComponent<Renderer>().material.color
                                                                          Color.grey
                                                                                        &
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.green
                                                                                        &
pin3menos.GetComponent<Renderer>().material.color
                                                                      Color.grey
                                                                                        &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.grey;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.grey;
      distoprig.GetComponent<Renderer>().material.color = Color.red;
      distop.GetComponent<Renderer>().material.color = Color.red;
    }else
    //3
         (pinmenossig.GetComponent<Renderer>().material.color
                                                                         Color.green
                                                                                        &
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.green
                                                                                        &
pin3menos.GetComponent<Renderer>().material.color
                                                                      Color.grey
                                                                                        &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.grey;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.red;
    }else
    //4
```

```
(pinmenossig.GetComponent<Renderer>().material.color
                                                                         Color.grey
                                                                                       &
pin2menos.GetComponent<Renderer>().material.color
                                                                                       &
                                                                     Color.grey
pin3menos.GetComponent<Renderer>().material.color
                                                           ==
                                                                     Color.green
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
    {
      disbot.GetComponent<Renderer>().material.color = Color.grey;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }else
    //5
    if
         (pinmenossig.GetComponent<Renderer>().material.color
                                                                        Color.green
                                                                                       &
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.grey
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                     Color.green
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.grey;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.red;
    }else
    //6
         (pinmenossig.GetComponent<Renderer>().material.color
                                                                         Color.grey
                                                                                       &
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.green
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                     Color.green
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
    {
      disbot.GetComponent<Renderer>().material.color = Color.grey;
```

```
disbotleft.GetComponent<Renderer>().material.color = Color.grey;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.red;
      distop.GetComponent<Renderer>().material.color = Color.red;
    }else
    //7
    if
         (pinmenossig.GetComponent<Renderer>().material.color
                                                                         Color.green
                                                                                        &
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.green
                                                                                        &
pin3menos.GetComponent<Renderer>().material.color
                                                                     Color.green
                                                                                        &
pinmassig.GetComponent<Renderer>().material.color == Color.grey)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.grey;
      dismid.GetComponent<Renderer>().material.color = Color.grey;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }else
    //8
    if
         (pinmenossig.GetComponent<Renderer>().material.color
                                                                          Color.grey
                                                                                        &
pin2menos.GetComponent<Renderer>().material.color
                                                            ==
                                                                      Color.grey
                                                                                        &
pin3menos.GetComponent<Renderer>().material.color
                                                                      Color.grey
                                                                                        &
                                                            ==
pinmassig.GetComponent<Renderer>().material.color == Color.green)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.red;
      distop.GetComponent<Renderer>().material.color = Color.red;
```

```
}
   //9
   else if (pinmenossig.GetComponent<Renderer>().material.color == Color.green
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.grey
                                                                                      &
pin3menos.GetComponent<Renderer>().material.color
                                                                     Color.grey
                                                                                      &
pinmassig.GetComponent<Renderer>().material.color == Color.green)
   {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
   }
   //10
   else if (pinmenossig.GetComponent<Renderer>().material.color == Color.grey
                                                                                      &
pin2menos.GetComponent<Renderer>().material.color
                                                                    Color.green
                                                                                      &
pin3menos.GetComponent<Renderer>().material.color
                                                                     Color.grey
                                                                                      &
pinmassig.GetComponent<Renderer>().material.color == Color.green)
   {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
   }
   //11
   else if (pinmenossig.GetComponent<Renderer>().material.color == Color.green
pin2menos.GetComponent<Renderer>().material.color
                                                                                      &
                                                                    Color.green
```

```
pin3menos.GetComponent<Renderer>().material.color
                                                                                       &
                                                                     Color.grey
pinmassig.GetComponent<Renderer>().material.color == Color.green)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }
    //12
             (pinmenossig.GetComponent<Renderer>().material.color == Color.grey
    else if
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.grey
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                    Color.green
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.green)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }
    //13
    else if (pinmenossig.GetComponent<Renderer>().material.color == Color.green
pin2menos.GetComponent<Renderer>().material.color
                                                                     Color.grey
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                    Color.green
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.green)
    {
```

```
disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }
    //14
    else if (pinmenossig.GetComponent<Renderer>().material.color == Color.grey
                                                                                       &
pin2menos.GetComponent<Renderer>().material.color
                                                                    Color.green
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                    Color.green
                                                                                       &
pinmassig.GetComponent<Renderer>().material.color == Color.green)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
      distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }
    //15
    else if (pinmenossig.GetComponent<Renderer>().material.color == Color.green
                                                                                      &
pin2menos.GetComponent<Renderer>().material.color
                                                                    Color.green
                                                                                       &
pin3menos.GetComponent<Renderer>().material.color
                                                                    Color.green
                                                                                       &
                                                          ==
pinmassig.GetComponent<Renderer>().material.color == Color.green)
    {
      disbot.GetComponent<Renderer>().material.color = Color.red;
      disbotleft.GetComponent<Renderer>().material.color = Color.red;
      disbotrig.GetComponent<Renderer>().material.color = Color.red;
      dismid.GetComponent<Renderer>().material.color = Color.red;
```

```
distopleft.GetComponent<Renderer>().material.color = Color.red;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }
  }
 // Está pendiente en todo momento
  void Update () {
       if(pinact .GetComponent<Renderer>().material.color == Color .green)
    {
      numero();
    }
    else
    {
      disbot.GetComponent<Renderer>().material.color = Color.grey;
      disbotleft.GetComponent<Renderer>().material.color = Color.grey;
      disbotrig.GetComponent<Renderer>().material.color = Color.grey;
      dismid.GetComponent<Renderer>().material.color = Color.grey;
      distopleft.GetComponent<Renderer>().material.color = Color.grey;
      distoprig.GetComponent<Renderer>().material.color = Color.grey;
      distop.GetComponent<Renderer>().material.color = Color.grey;
    }
       }
}
```

En este caso la clase "numero" almacena todas las posibilidades de las activaciones de los pines asignados a los bits. En la clase "Update", que es propia del programa Unity y siempre se está ejecutando, está atenta a los eventos de los ratos para activar el display adecuado.

Esta aplicación la pueden descargar para el sistema operativo Windows en: https://www.dropbox.com/sh/rxc4uwzyqsoilyd/AABISkpa98pdTjAwUXrlMRIea?dl=0

Carlos Bretón Timón Carlos Castro Pérez

Memoria

15 Software de aprendizaje para la enseñanza en la ingeniería

15.1 Objetivos

Queríamos relacionar la aplicación realizada con la asignatura "Informática Industrial".

Con la herramienta SIENA podremos ayudar a los alumnos en su aprendizaje y que

tengan la posibilidad de que realicen una autoevaluación de la asignatura. También

facilitará a los profesores realizar test evaluativos.

En este apartado se mostrará la elaboración del cuestionario en el programa SIENA.

15.2 ¿Qué es siena?

La herramienta SIENA (Sistema Integrado de Enseñanza y Aprendizaje) es una

aplicación web, creada por la Universidad de La Laguna, cuya función es detectar los

conocimientos previos de un alumno o como ayuda para el auto aprendizaje y la

autoevaluación.

SIENA está diseñado para trabajar con mapas conceptuales. El software usa estos mapas

conceptuales para representar los conocimientos de un tema concreto de una manera

organizada. Estos se elaboran utilizando el software compendium LD, este programa

permite exportar la información en formato XML en la herramienta SIENA.

Una vez realizado el mapa conceptual se le pueden asignar conceptos en forma de

preguntas. El alumno será evaluado por medio de estas preguntas. Estas preguntas son de

opción múltiple y tendrán unas características específicas.

• Grado de correlación entre la pregunta y el concepto.

• La dificultad de la pregunta.

• La respuesta correcta.

• Conjeturas. facilidad de responder a la pregunta solo teniendo en cuenta el factor

suerte.

• La estimación de los conocimientos previos.

• Tiempo de respuesta.

61

El motivo de estos parámetros es para que el software SIENA sea capaz de estimar correctamente la capacidad y la compresión de un concepto dado, es decir que el programa sea capaz de adaptarse al alumno.

15.3 Pasos a seguir

15.3.1 Registrarse en SIENA

Lo primero que hay que hacer es registrarse en SIENA. Dentro de este programa se distinguen tres tipos de usuarios: administrador, maestro y estudiante. En nuestro caso nos registramos como administrador. La función del administrador es agregar nuevos temas, activar a los usuarios, tanto profesores como alumnos. También puede ver, editar y eliminar preguntas guardadas y nodos.

Para registrarse hay que acceder a la página oficial de SIENA: http://sienasocial.ull.es/

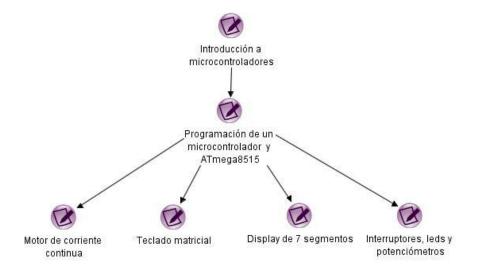
Hasta que alguno de los administradores autorice el registro no es posible trabajar en SIENA.

15.3.2 CompendiumLD

Para el mapa conceptual de SIENA es necesario elaborar nodos, donde cada uno de éstos represente los conceptos que se quieren tratar en la asignatura. El mapa conceptual se elabora con el programa CompendiumLD.

CompendiumLD es una herramienta de software para el diseño de actividades de aprendizaje utilizando una interfaz visual flexible. Es una herramienta de apoyo para profesores y otras personas involucradas en la educación para ayudarles a expresar sus ideas y trazas la secuencia de diseño o de aprendizaje.

Para crear un nuevo mapa conceptual se crea un nuevo proyecto. Luego para crear los nodos, hay que seleccionar el icono "*Note Node*" haciendo click con el botón izquierdo del ratón, y sin soltarlo, deslizarlo hasta el mapa de trabajo abierto a la derecha. El mapa conceptual queda de la siguiente forma:



15-1. Mapa Conceptual

Una vez creado el mapa, se exporta en formato XML para subirlo a la plataforma de SIENA.

15.3.3 Elaborar el cuestionario

Una vez registrados en SIENA se procede a la elaboración del cuestionario para la asignatura. Lo primero que se debe hacer es elaborar un mapa conceptual de los conceptos que se van a tratar en la asignatura para los alumnos.

Preparar una asignatura para SIENA significa establecer un mapa conceptual jerárquico. Una vez diseñado el mapa, hay que preparar contenidos de información y preguntas para cada nodo al objeto de lograr un desempeño efectivo del alumno en cada uno de ellos. SIENA se encargará de realizar los test desde los nodos inferiores o más simples a los superiores o más complejos.

Una vez elaborado el mapa conceptual, se procede a la elaboración del cuestionario. Para ello hay que acceder a SIENA, como se mencionó anteriormente. Luego hay que matricularse y acceder a la asignatura para la que se realiza el cuestionario, en este caso "Informática Industrial" (nombre dado en el programa SIENA). Se selecciona "nueva Asignatura" y a continuación aparecerá el siguiente recuadro:



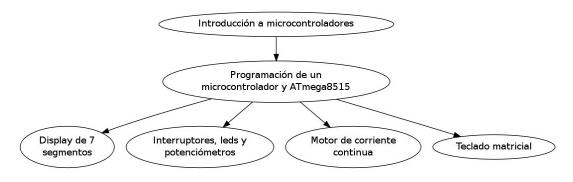
15-2. Crear asignatura

Accediendo a lista de nodos, se importa el mapa conceptual creado en el programa CompendiumLD:



15-3. Importar mapa conceptual

Cuando esté importado se genera automáticamente una imagen del mapa conceptual y un listado de nodos.



15-4. Imagen mapa conceptual

Lista de nodos de la asignatura Informatica industrial Nombre Ver Editar Borrar 7 Display de 7 segmentos Ver Editar Borrar Editar Borrar Interruptores, leds y potenciómetros Ver Introducción a microcontroladores Editar Borrar Motor de corriente continua Editar Borrar Ver Teclado matricial Ver Borrar Programación de un microcontrolador y ATmega8515 Ver Editar Borrar 1-6 / 6

15-5. Listado de nodos

Una vez completado estos pasos, ya se puede proceder a elaborar las preguntas para el cuestionario, esto se hace accediendo a la sección "lista de preguntas de la asignatura". Se selecciona en la parte inferior de la página el enlace Nueva pregunta.



15-6. Nueva pregunta

La pregunta se redacta en la casilla "Contenido". El tiempo de respuesta que tienen los alumnos para la pregunta se mide en segundos. En la respuesta correcta se debe incluir el número de la respuesta empezando por cero, es decir, si la pregunta correcta es la tercera en esta casilla se pondrá el número dos.

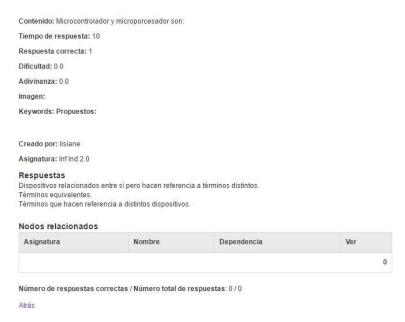
En "dificultad "se añade un número comprendido en entre 0 y 1, donde el 0 significa poco grado de dificultad y 1 mayor grado de dificultad. La casilla adivinanza representa la probabilidad de acertar en caso de no saber la respuesta. El rango es de 0 a 1. También SIENA permite la posibilidad de insertar un archivo a la pregunta.

Una vez terminada una pregunta hay que asociarla con las respuestas.



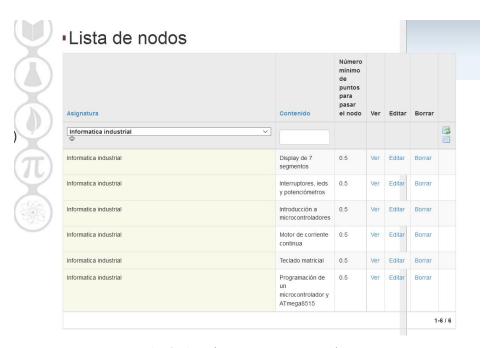
15-7. Relacionar pregunta con respuestas

A continuación, se muestra una de las preguntas elaboradas para el cuestionario:



15-8. Pregunta finalizada

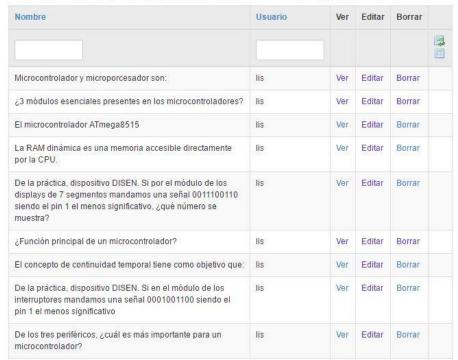
Una vez terminada la pregunta, ésta se ha de relacionar con algún nodo de la asignatura. Esto se hace accediendo al enlace, que se encuentra en la parte inferior de la pregunta, asociar nuevo nodo de la pregunta. Cuando se accede al enlace, aparece una lista con todos los nodos que se han creado en SIENA. Para acceder a los nodos de una asignatura sólo hay que escribir el nombre de ésta, y luego seleccionar la opción filtrar, esto permite ver los nodos que hay en ella.



15-9. Asociar pregunta a un nodo

Cuando la pregunta esté asociada al nodo aparecerá en el listado de preguntas de éste. Para ver cuántas preguntas se han elaborado para la asignatura, hay que dirigirse a la opción "Lista de Preguntas" y en ella aparecerá todas las preguntas que se han elaborado, permitiendo ver, editar o borrar cualquiera de éstas.

Lista de preguntas de la asignatura



15-10. Lista de preguntas

16 Presupuesto

Tabla 11. Presupuesto

Material	Precio €
Ordenador Portátil ACER AMD-FX7500 Radeon R7	769.00
Teclado y Ratón Externo Logitech	37.65
Software BLENDER ¹	0.00
Software UNITY ¹	0.00
Mano de Obra (15€/hora)	1800.00
Total	2606.65

17 Conclusión

La simulación es una técnica que ha sido empleada para la resolución de problemas. Los modelos de simulación son una abstracción de un sistema, que nos ayudan a tomar decisiones sobre el mismo sistema o, como en nuestro caso, sirven para mostrar el sistema de una manera más sencilla y didáctica.

A lo largo de este trabajo se han descrito una serie de elementos importantes que forman parte de la elaboración de una simulación, empezando desde los pasos a seguir en teoría hasta la implementación en el ordenador realizando los modelos 3D y finalizando con la aplicación terminada. El objetivo era crear un instrumento que sea de ayuda para los estudiantes de "Informática Industrial" en el momento de realizar las practicas.

Con este objetivo en mente se inició una investigación de lo que es la simulación ¿cómo se diseña?, ¿qué pasos se siguen para diseñarla?, ¿cómo se implementa?, ¿cuál es la función ?, ¿información que se pretende mostrar y de qué modo mostrarla?

Teniendo en cuenta los puntos señalados anteriormente, cumplimos exitosamente con los objetivos marcados para la simulación. Pensamos que fue acertada la selección de los programas. Como primera intensión, teníamos planeado usar Inventor como software, pero después de realizar el estudio lo desechamos porque la simulación en Inventor está

¹ Software Libre

dedicada para piezas mecánicas. En nuestro caso los modelos son componentes electrónicos e Inventor solo permite en sus simulaciones mover o girar las piezas, mientras que usando el Unity 3D fuimos capaces de mostrar los circuitos eléctricos o encender y apagar los componentes.

También mencionar el software educativo SIENA. Podemos concluir que es una herramienta informática de gran ayuda educativa. Permite a los alumnos el estudio de una forma autónoma logrando profundizar sus conocimientos en un tema determinado.

Esperamos que este proyecto ayude tanto a los profesores como a los alumnos a exponer todo el contenido que se da en la asignatura de "Informática Industrial" referente a los microcontroladores y a las practicas que se realizan.

18 Conclusion

Simulation is a technique that has been used for problem resolution. Simulation models are an abstraction of a system, that help us make decisions on the same system or, as in our case, serve as a didactic way for showing us the system more simply.

Throughout this work, we have described a number of important elements which form part of the development of a simulation, starting from the steps to follow in theory to implementation on the computer making the 3D models, for the purpose of making an application. The objective was to create a tool that is helpful for students of "Industrial Computing" when making the practices.

With this goal in mind an investigation about simulation was researched. How is it designed?, What steps are taken to design?, How is it implemented?, What is the function?, Information aims to show and how to show it?

Given the previous key points, successfully we fulfill the goals set for the simulation. We thought that was a good selection of programs. As first thought, we had planned to use Inventor software but after the study it was discarded, because the simulations in Inventor are dedicated to mechanical parts. In our case the models are electronic components and Inventor allowed only in their simulations moving or rotating the parts, while using the

Carlos Bretón Timón Carlos Castro Pérez

Memoria

Unity 3D we were able to show the electrical circuits or the components switching on and off.

Also mention the educational software SIENA, we can conclude that it is a software tool with high educational support. It allows students to study autonomously, improving their knowledge in a given subject.

We hope this project will help both teachers and students to explain all the content that is given regarding the subject of microcontrollers and the practices that are carried out in "Industrial Computing".

19 Bibliografía

 Datasheets de todos los componentes vistos en el microcontrolador y módulos

http://www.alldatasheet.com/datasheet-pdf/

- Tutorial del programa de modelado 3D Blender https://www.blender.org/support/tutorials/
- Tutorial del programa Unity3D
 https://unity3d.com/es/learn/tutorials

20 Referencias

- [1] "Página Oficial CATIA.".
- [2] "Página Oficial Inventor.".
- [3] "Página Oficial Blender.".
- [4] "Página Oficial Game Maker Studio.".
- [5] "Página Oficial Unity 3D.".



GRADO DE INGENIERÍA ELECTRONICA INDUSTRIAL Y AUTOMÁTICA

TRABAJO FIN DE GRADO

Titulo:

Propuesta de mejora de la asignatura "Informática Industrial" empleando la herramienta SIENA y simuladores 3D

ANEXO I

MODELOS 3D

Autores:

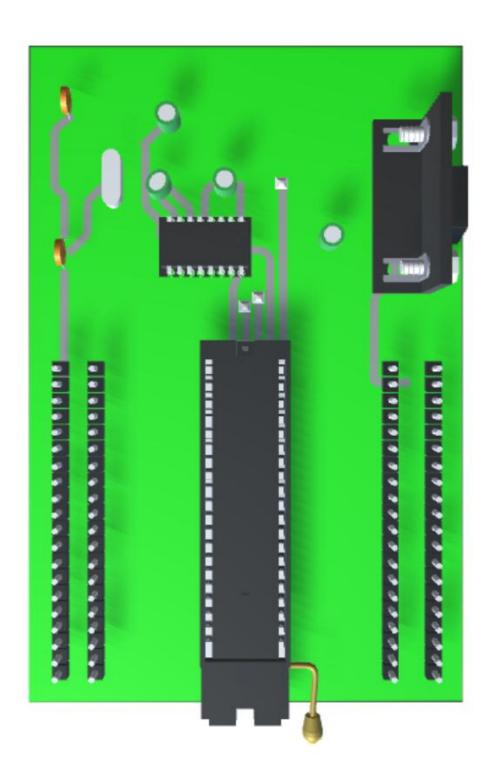
Carlos Castro Pérez

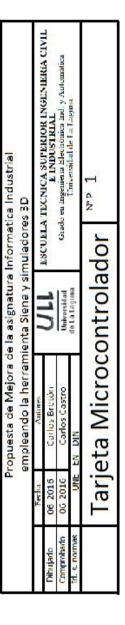
Carlos Bretón Timón

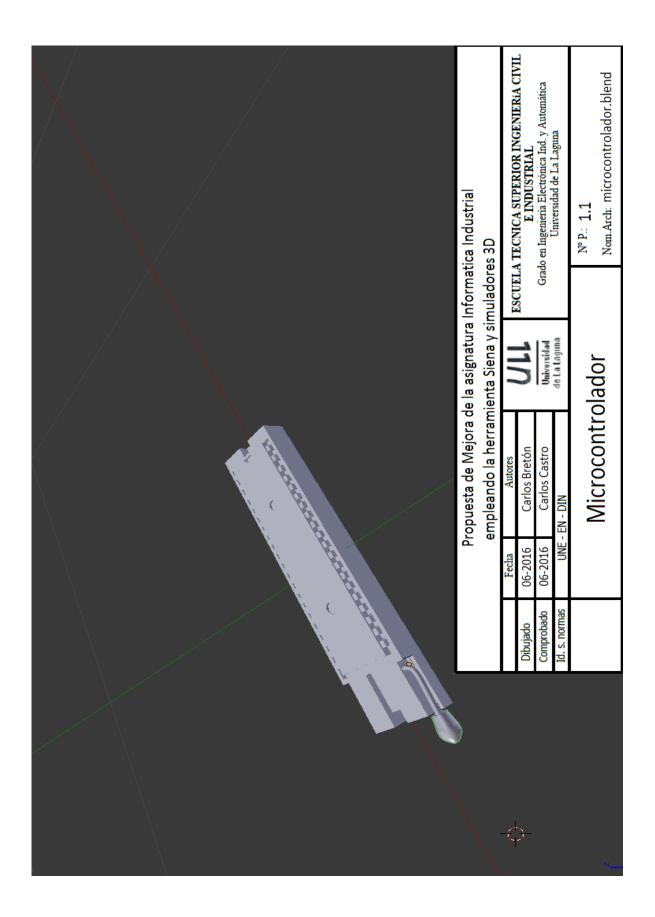
Tutor:

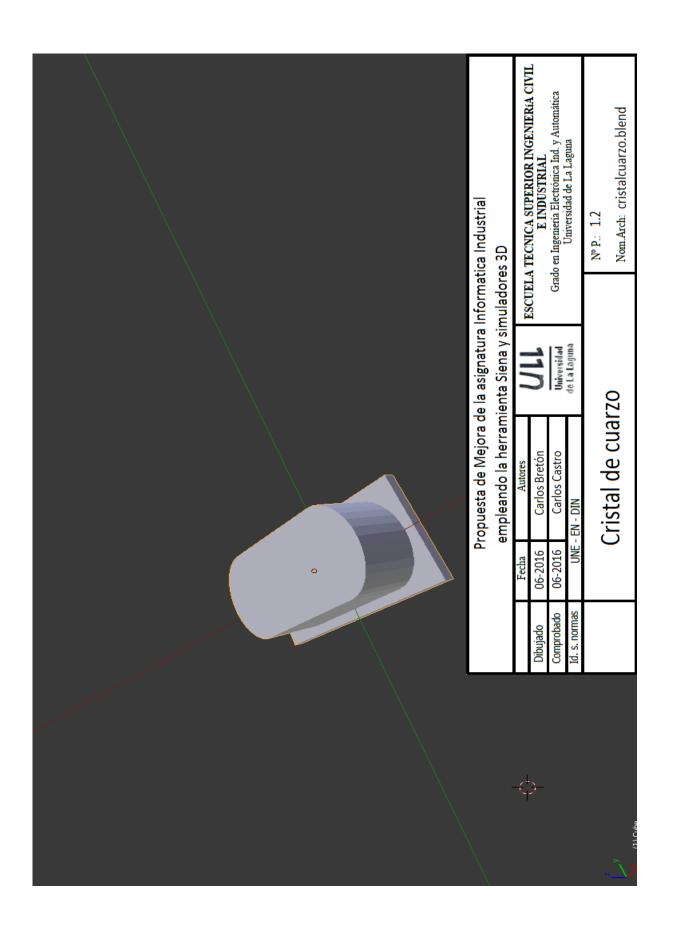
Evelio González González

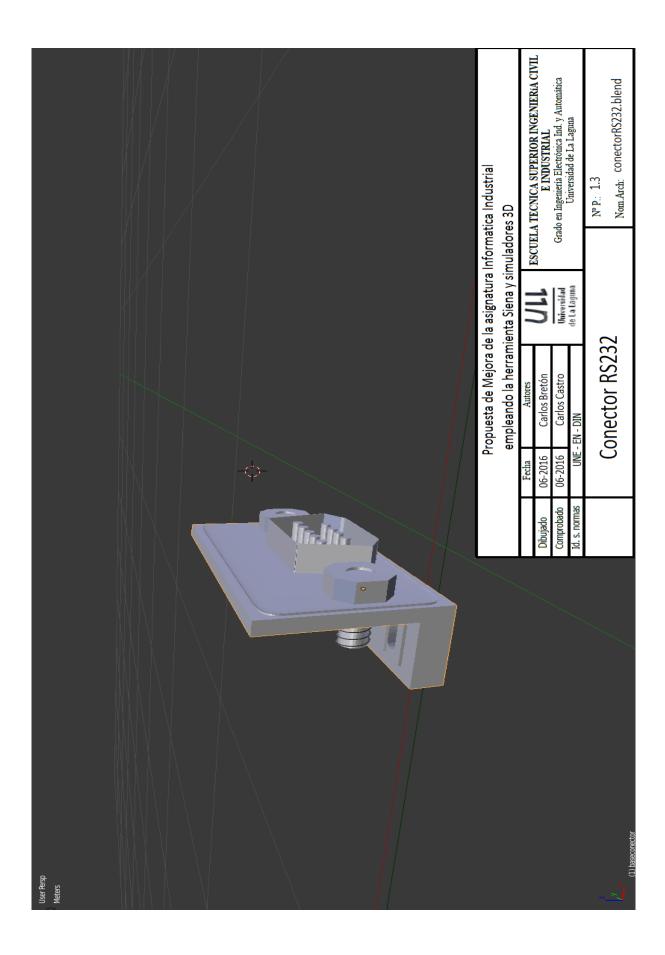
Septiembre 2016

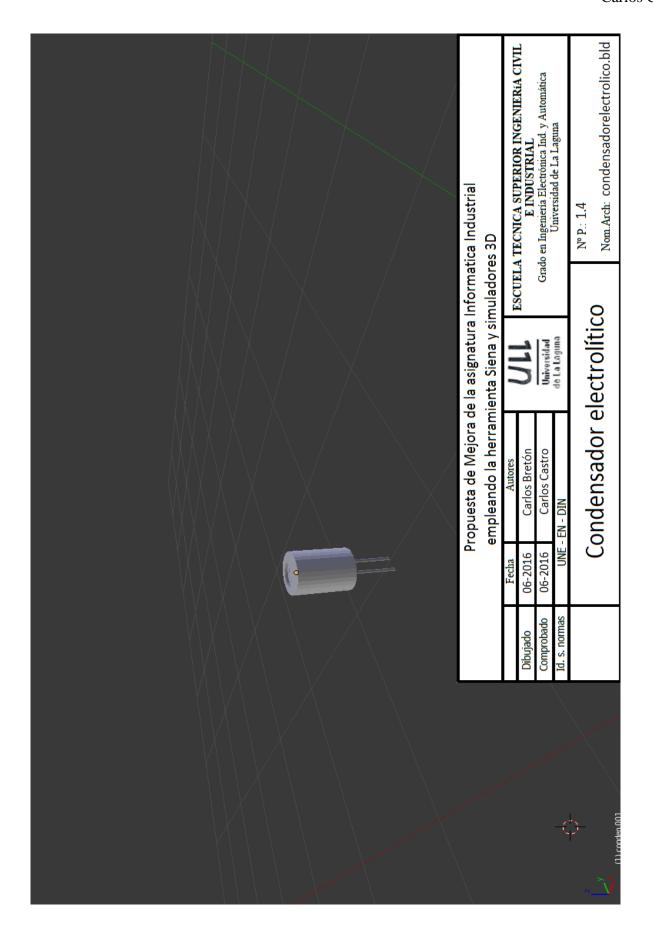


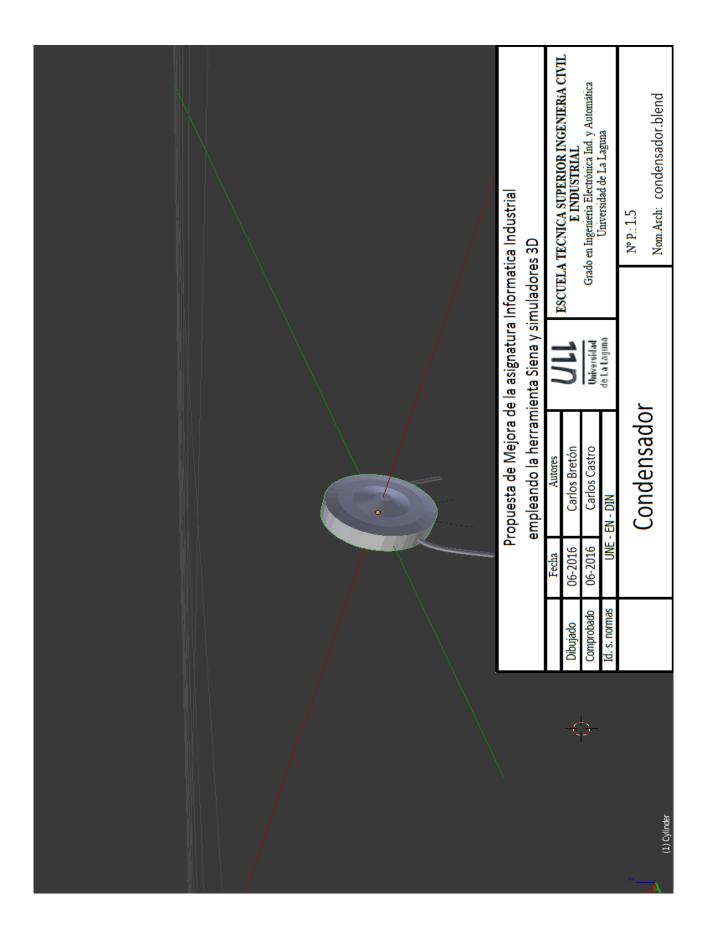


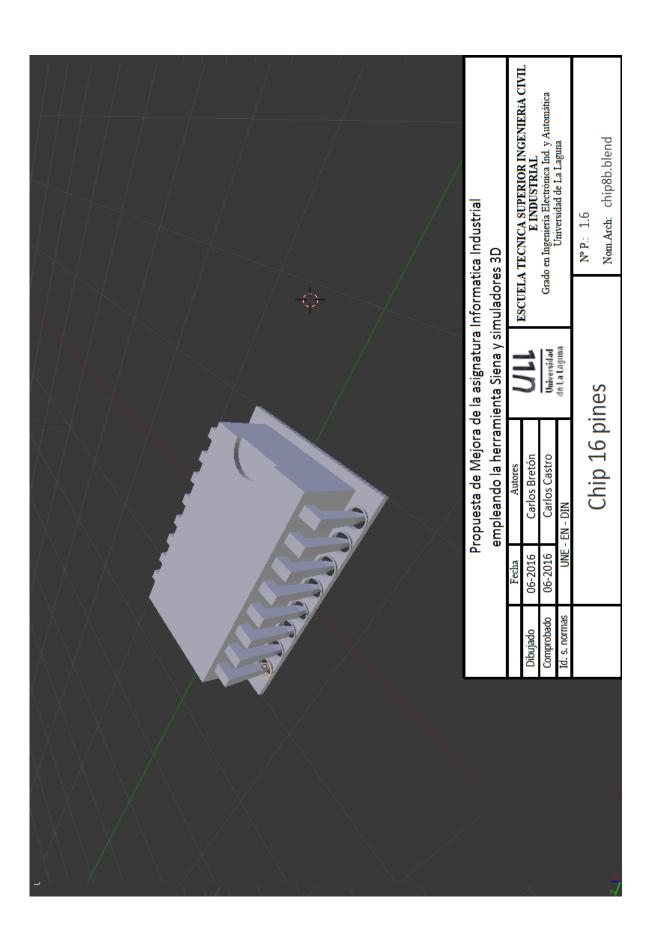


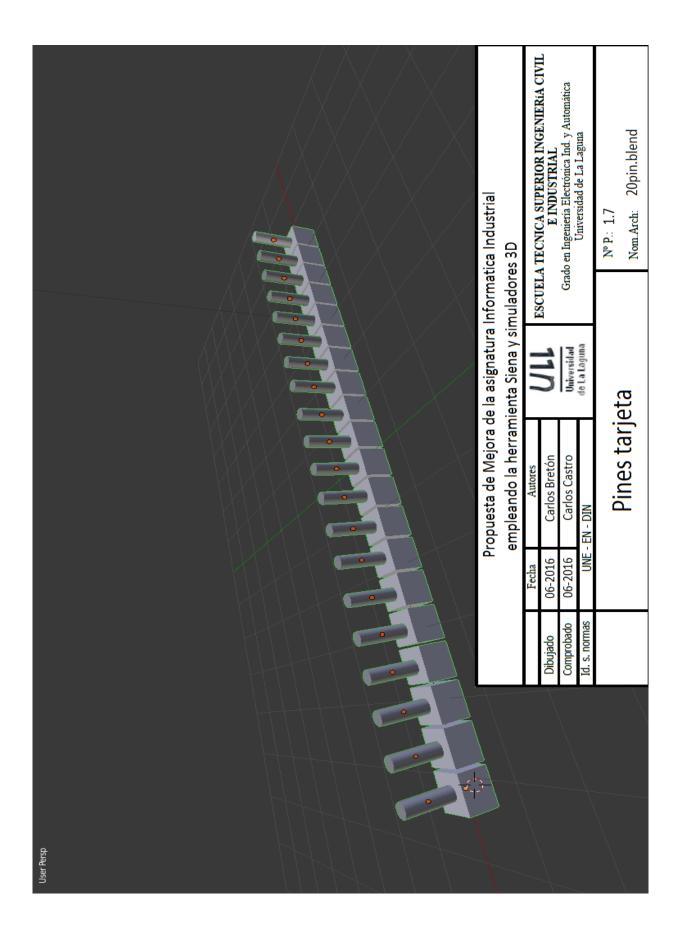


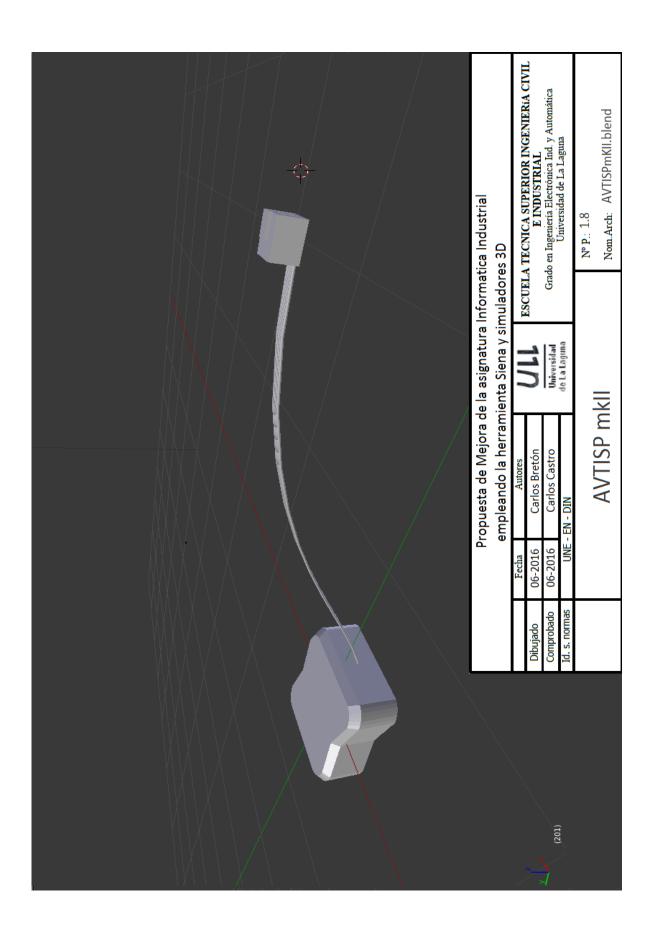


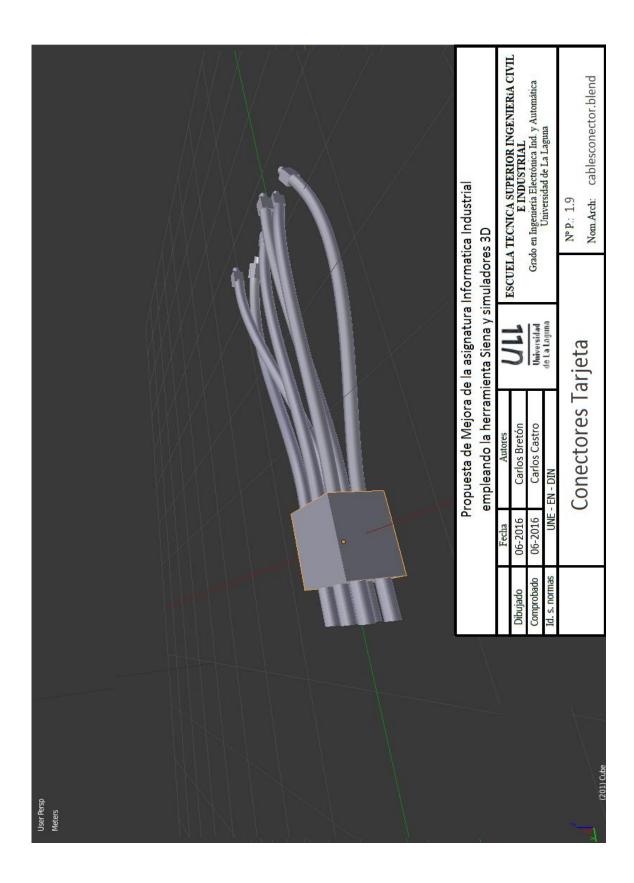


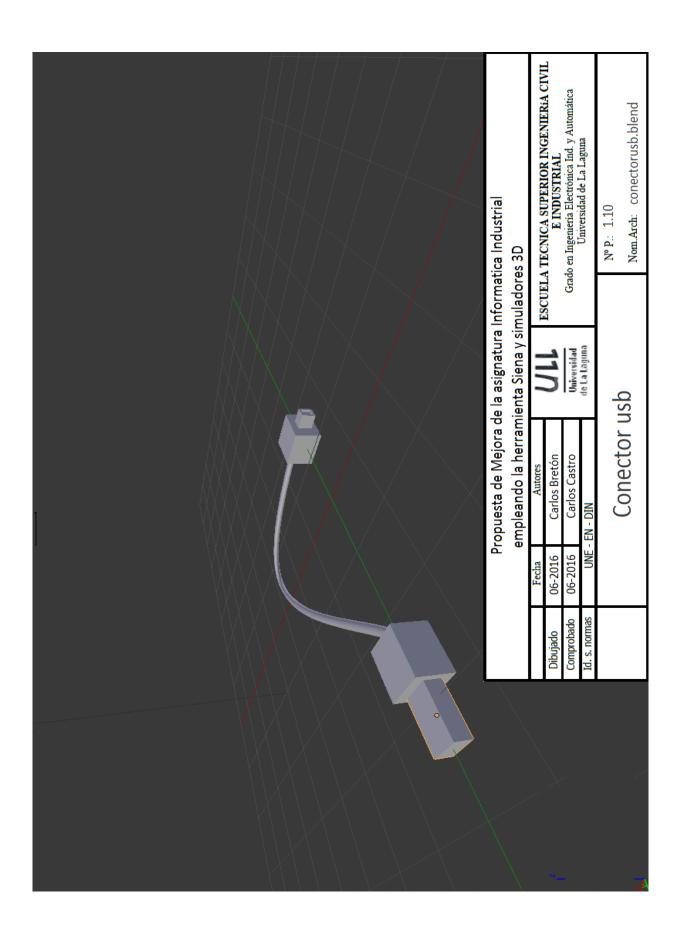


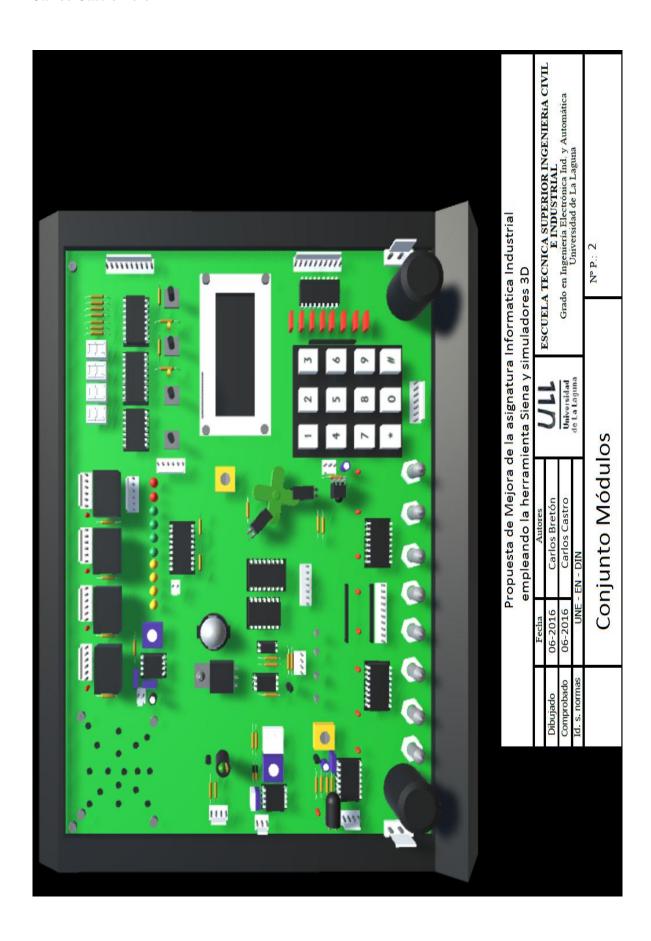


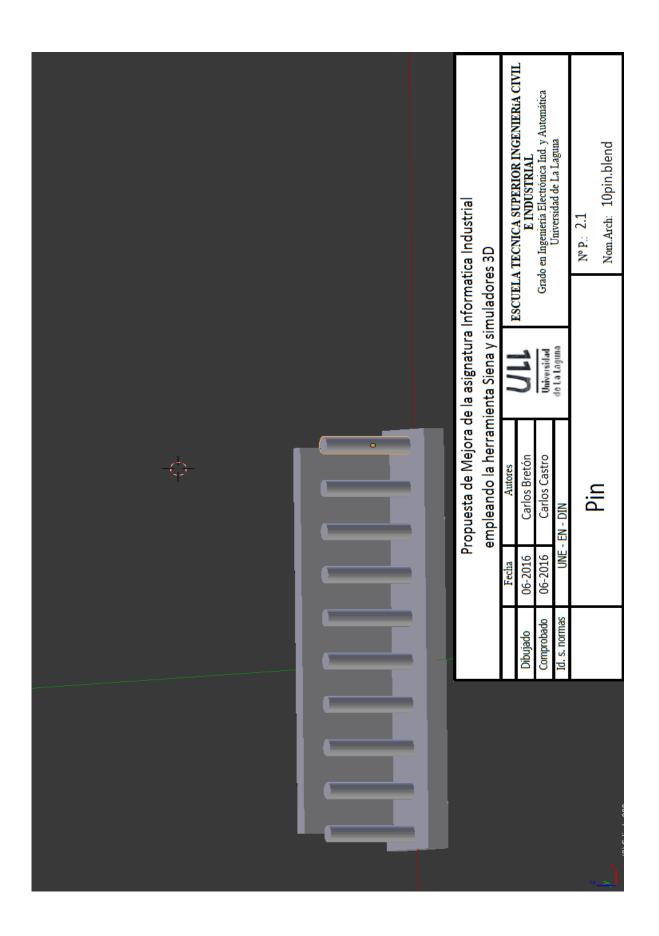


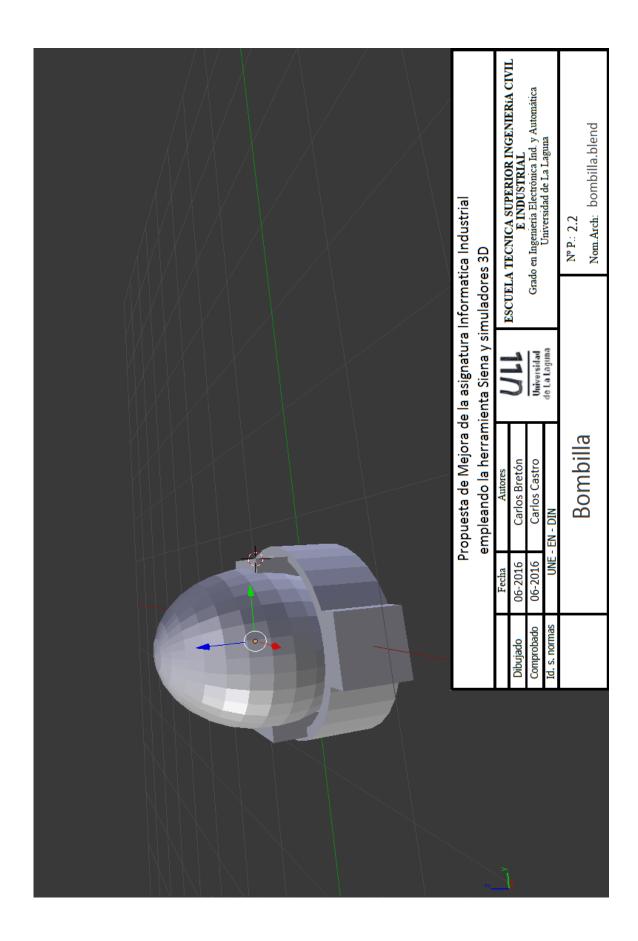


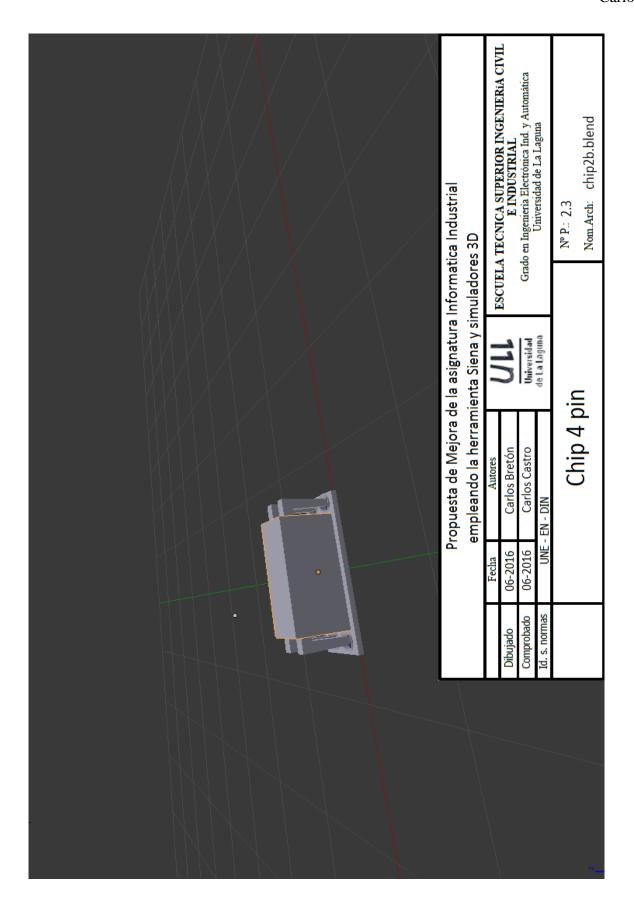


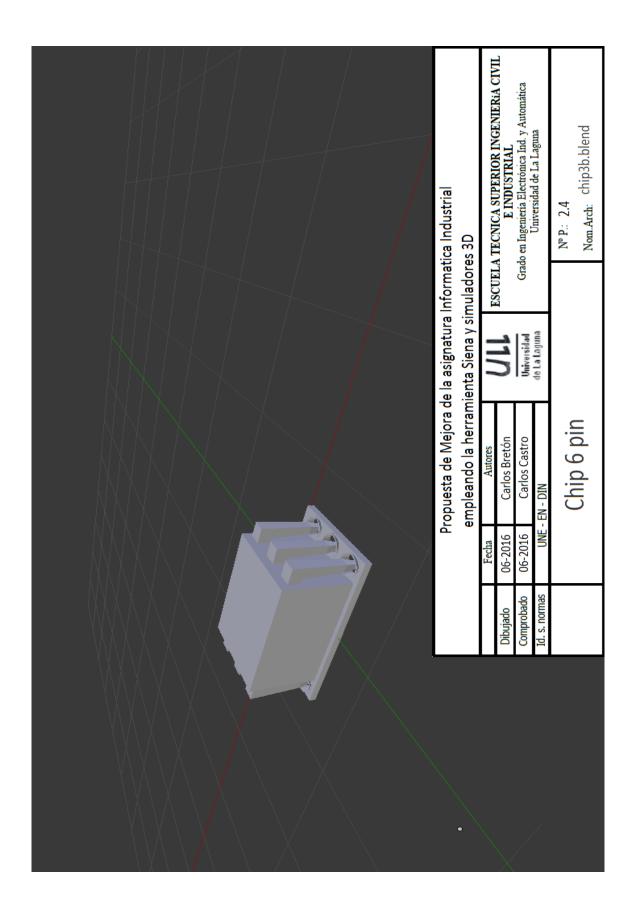


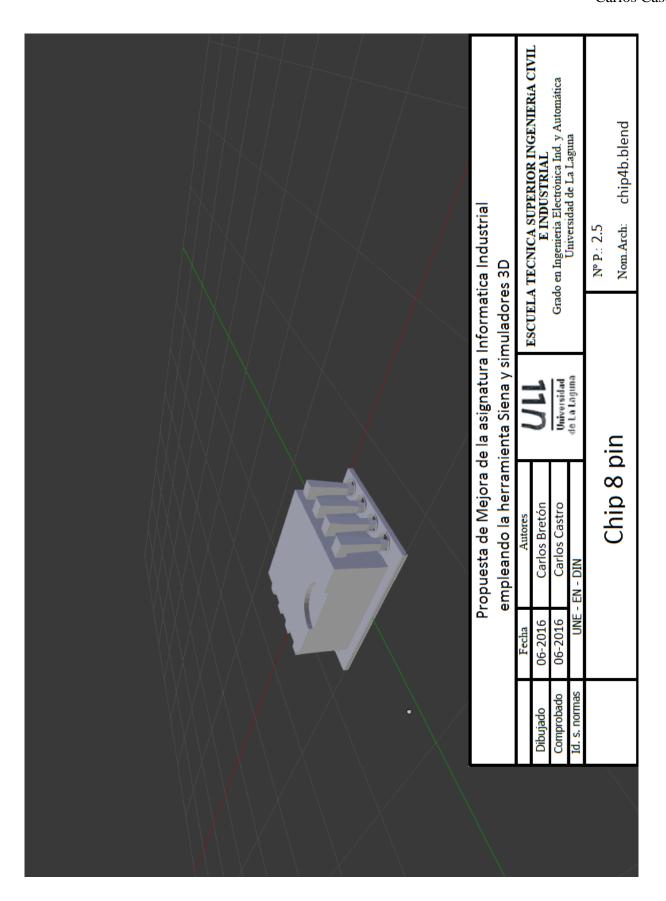


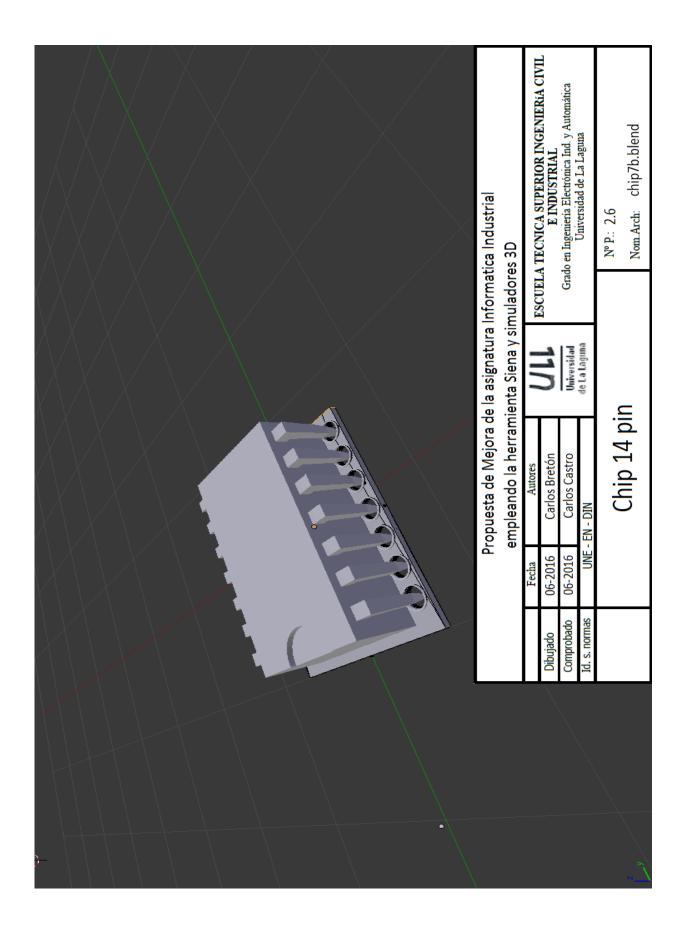


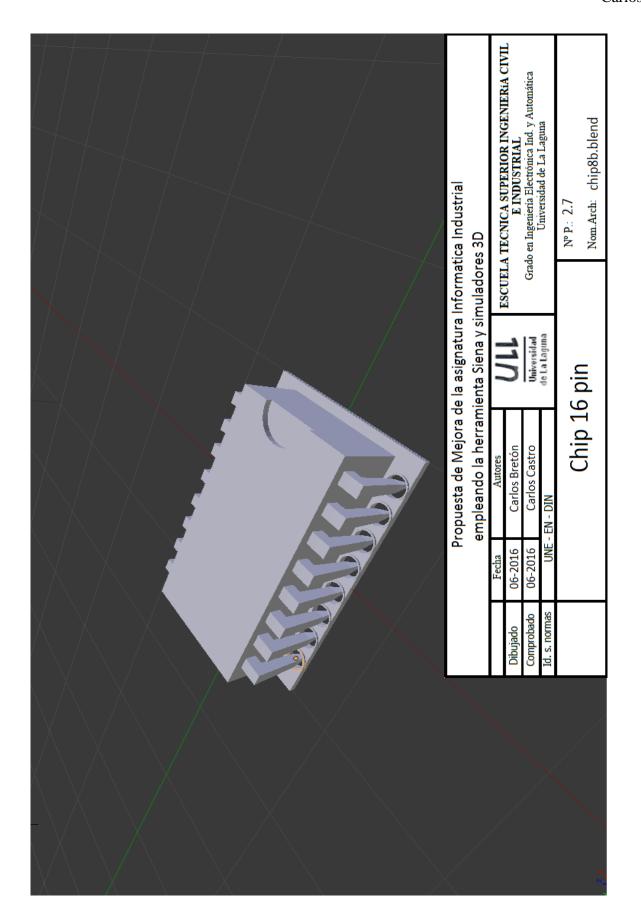


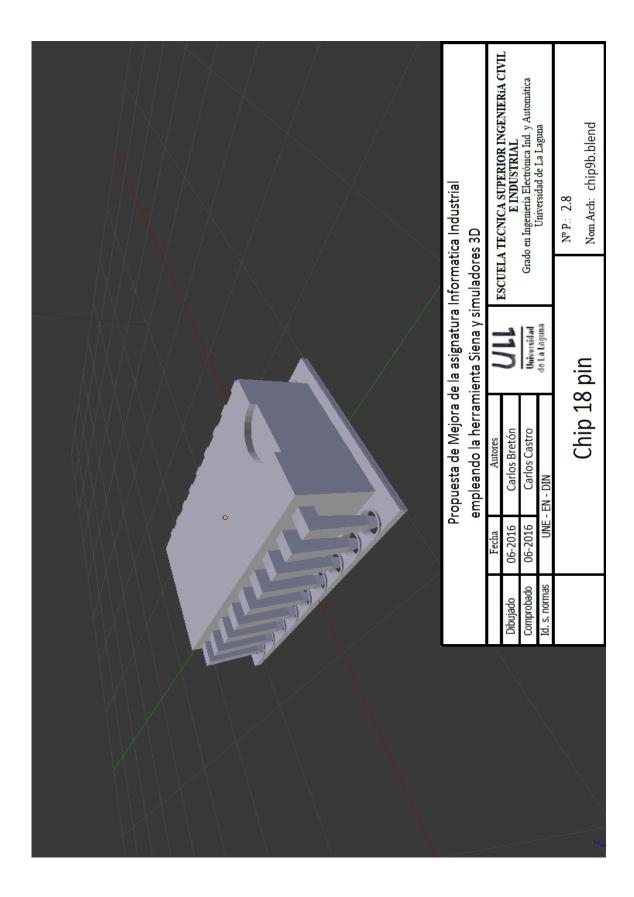


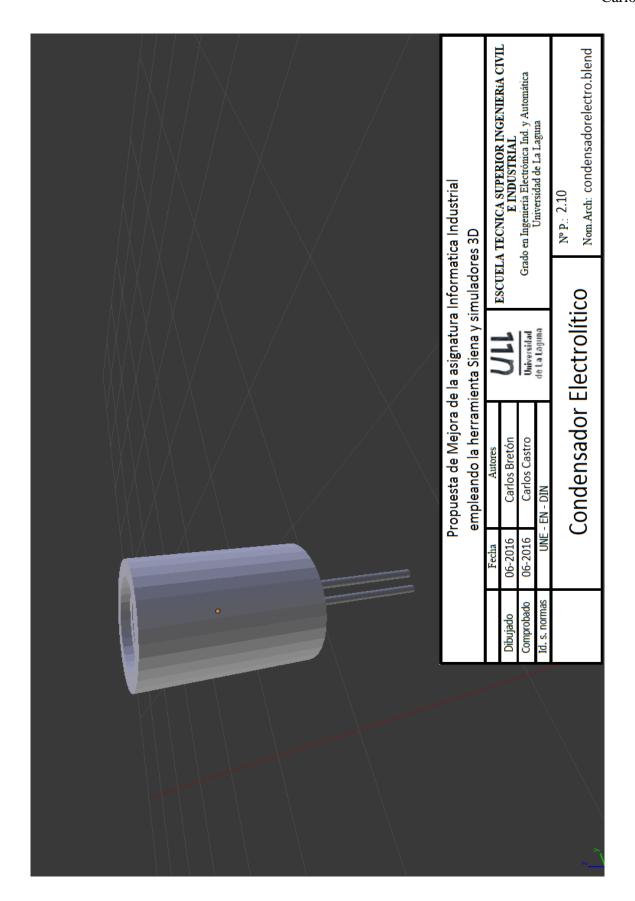


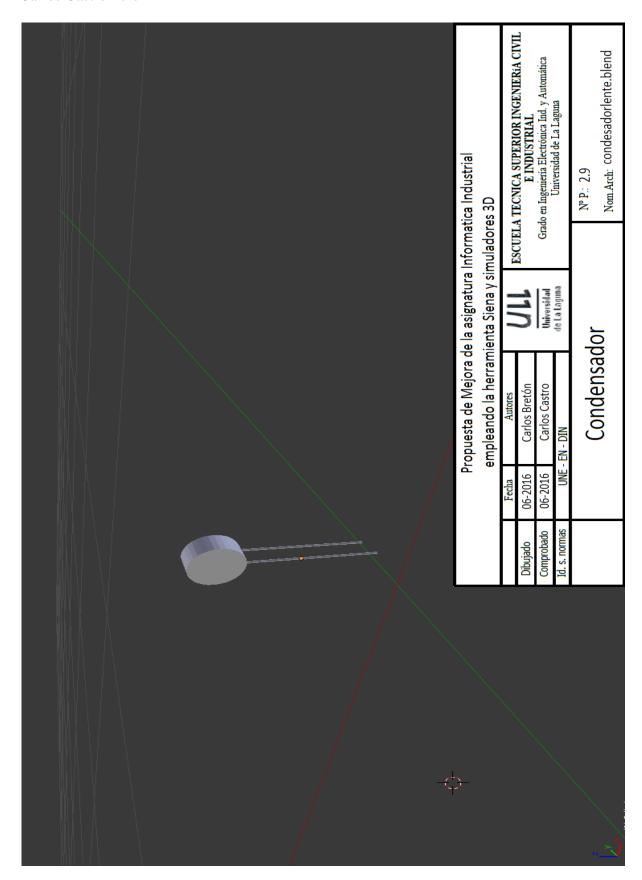


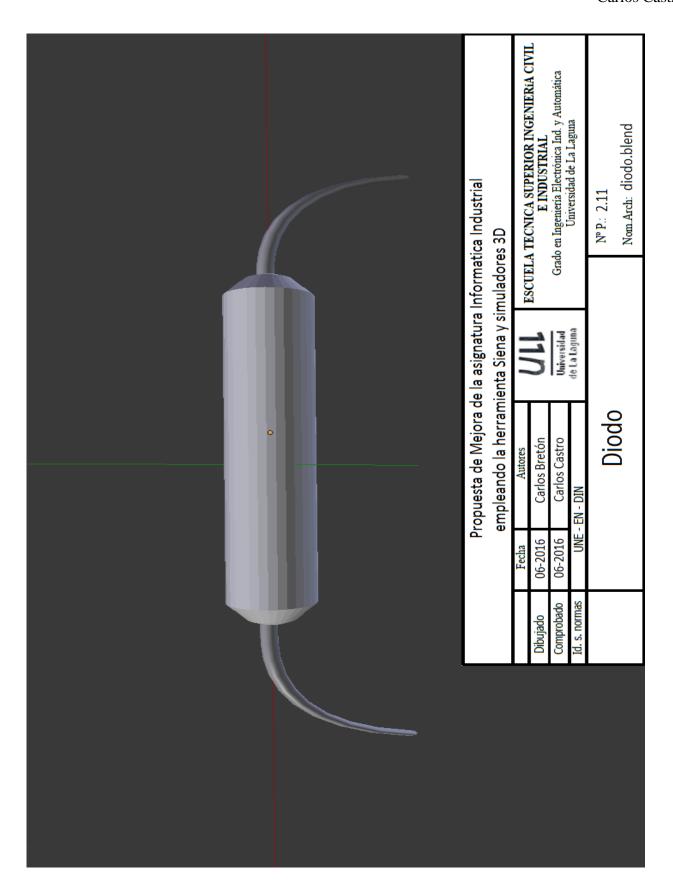


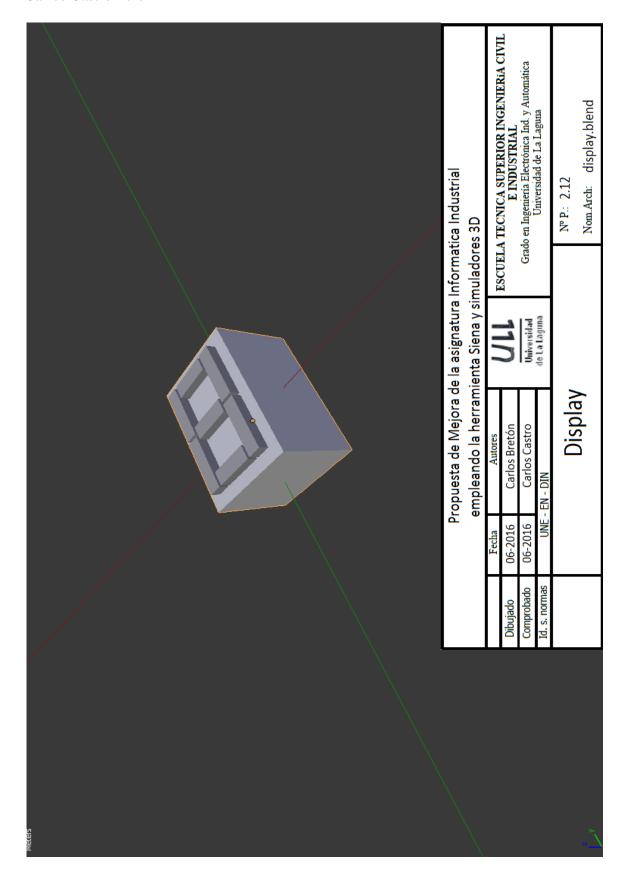


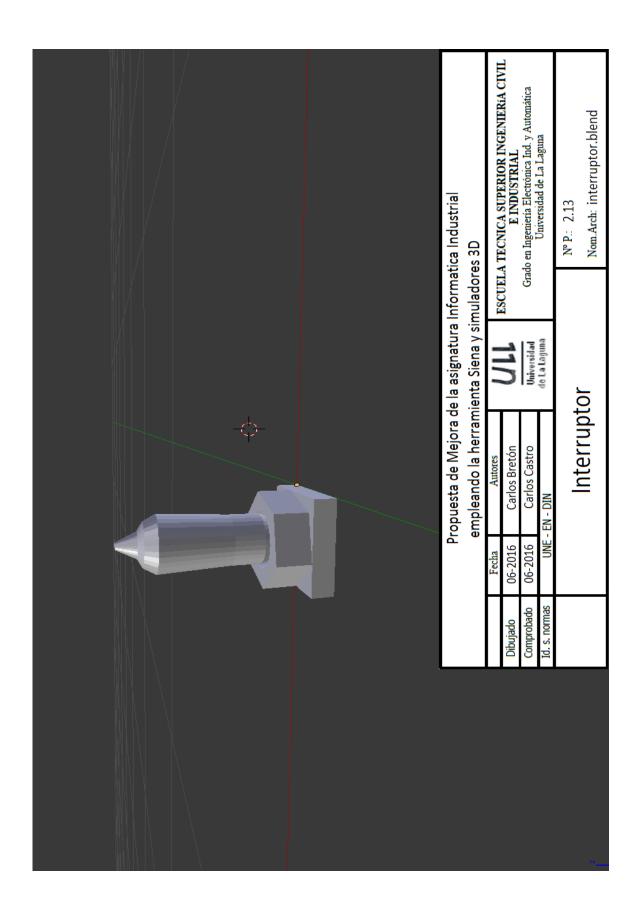


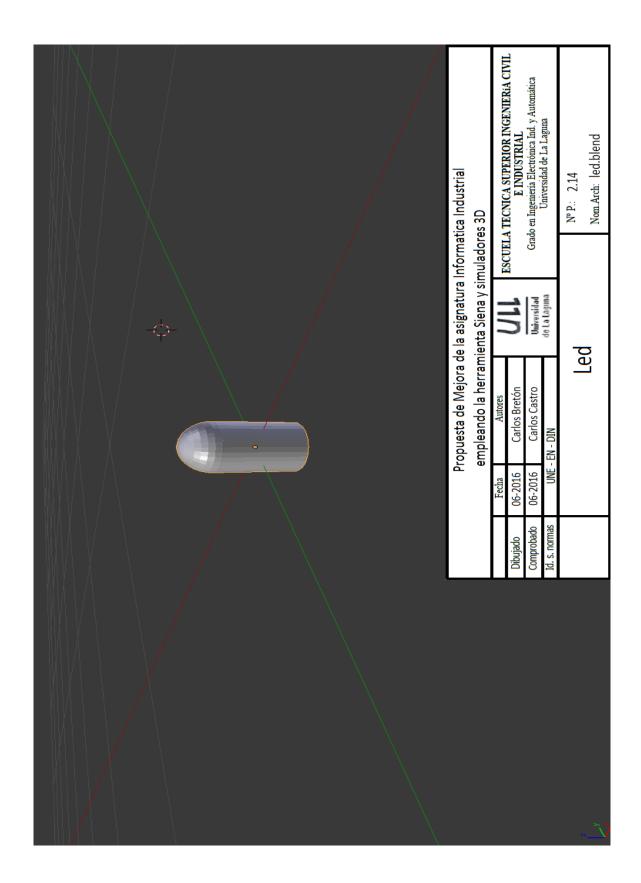


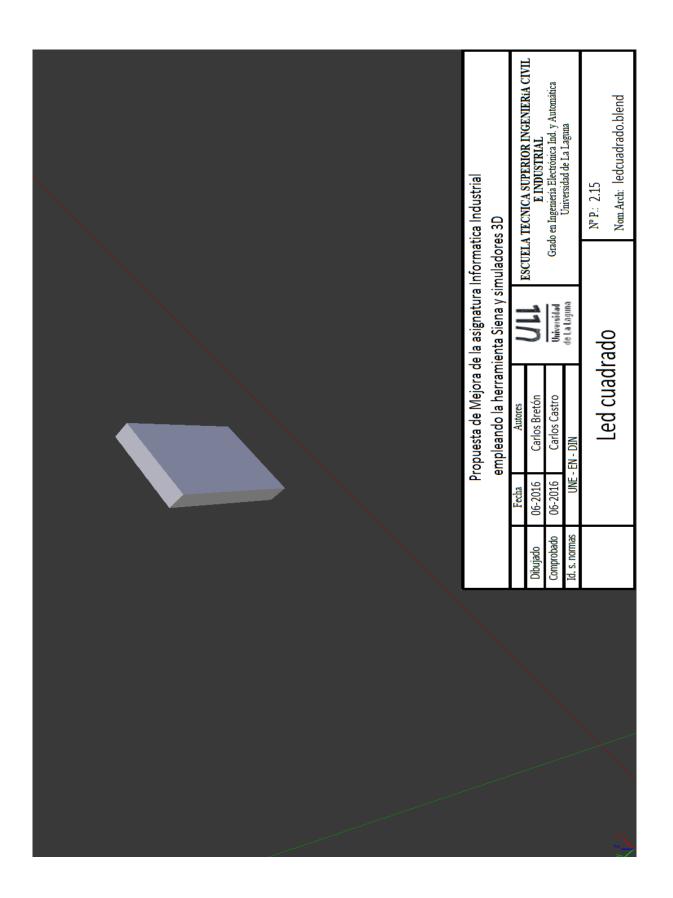


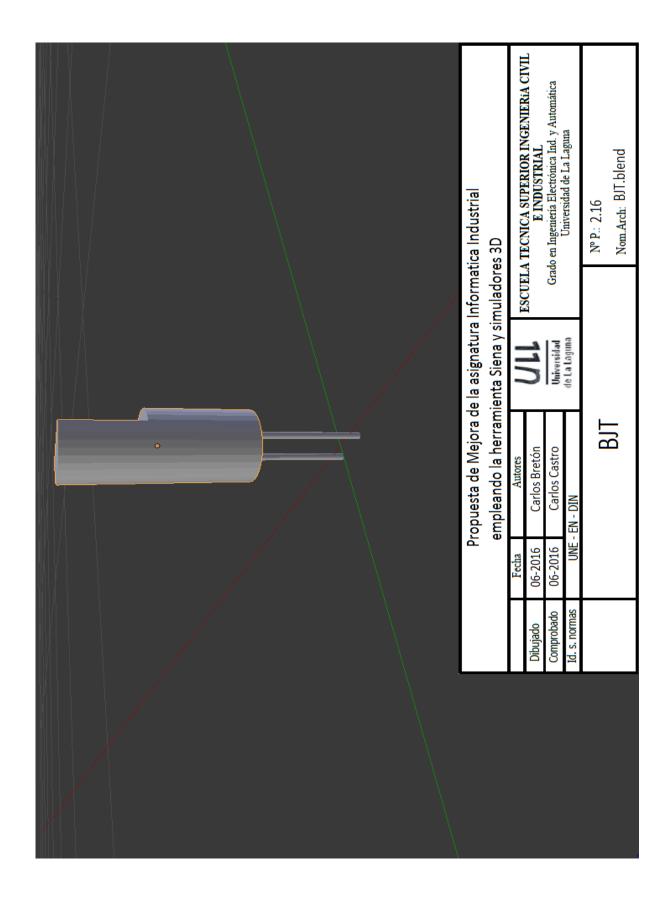


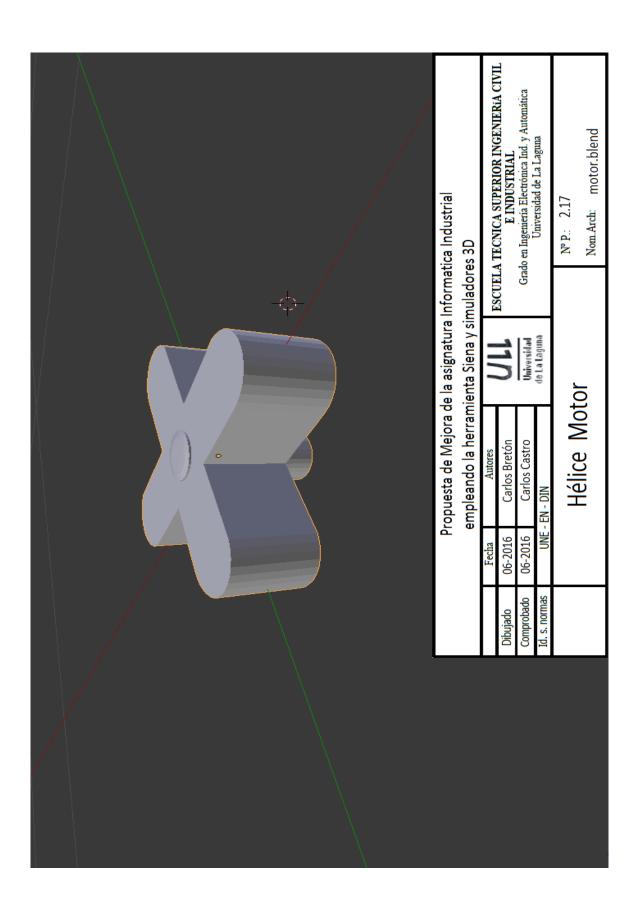


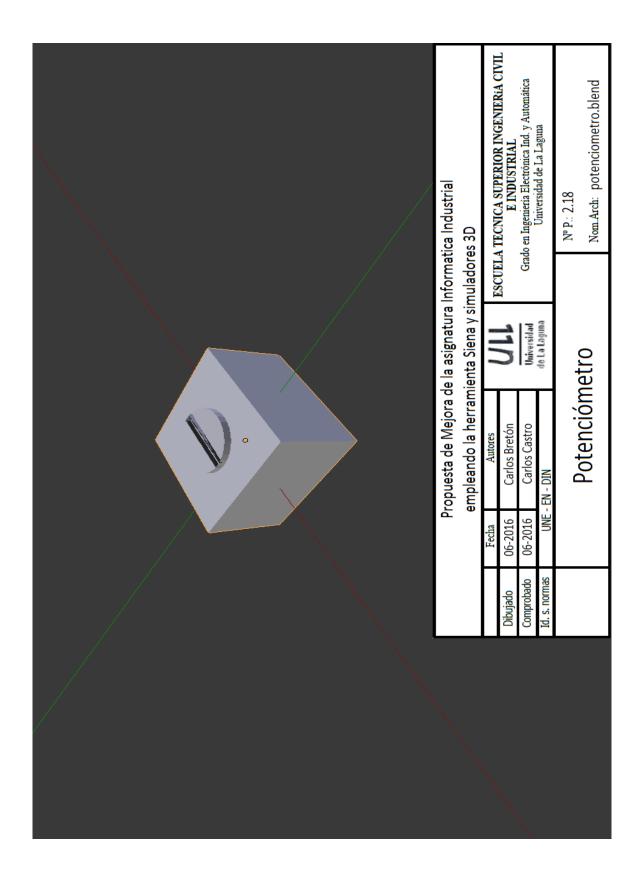


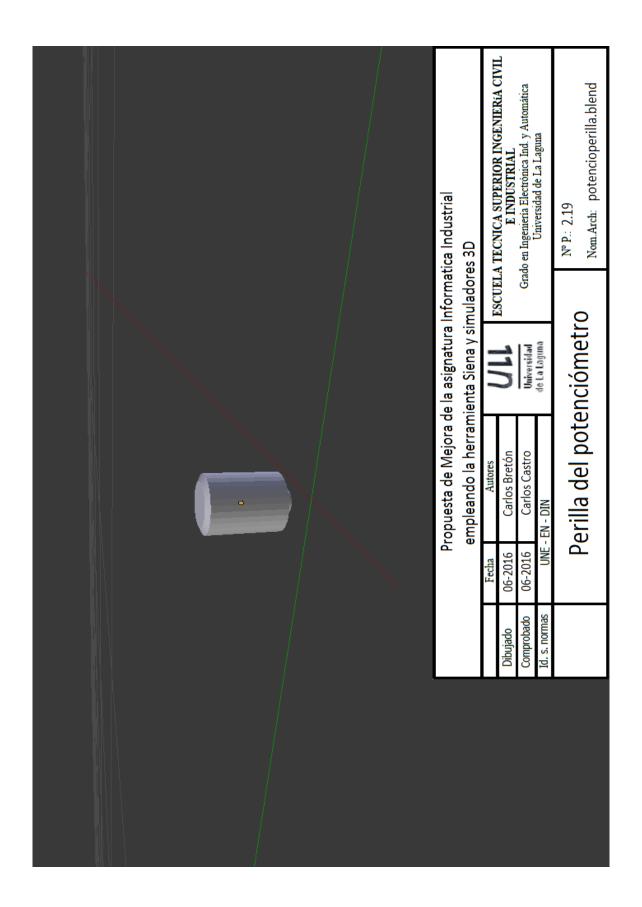


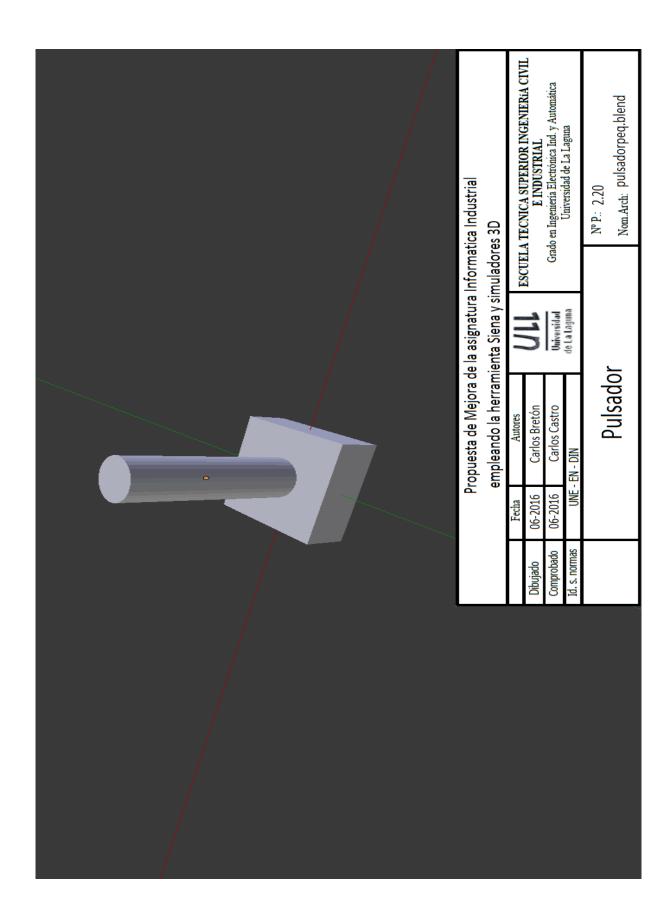


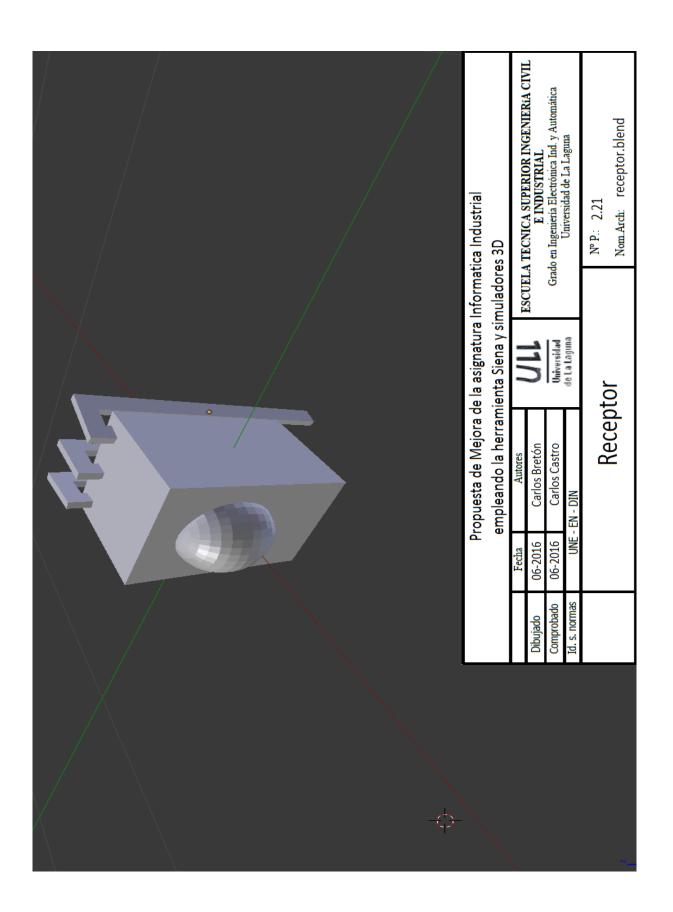


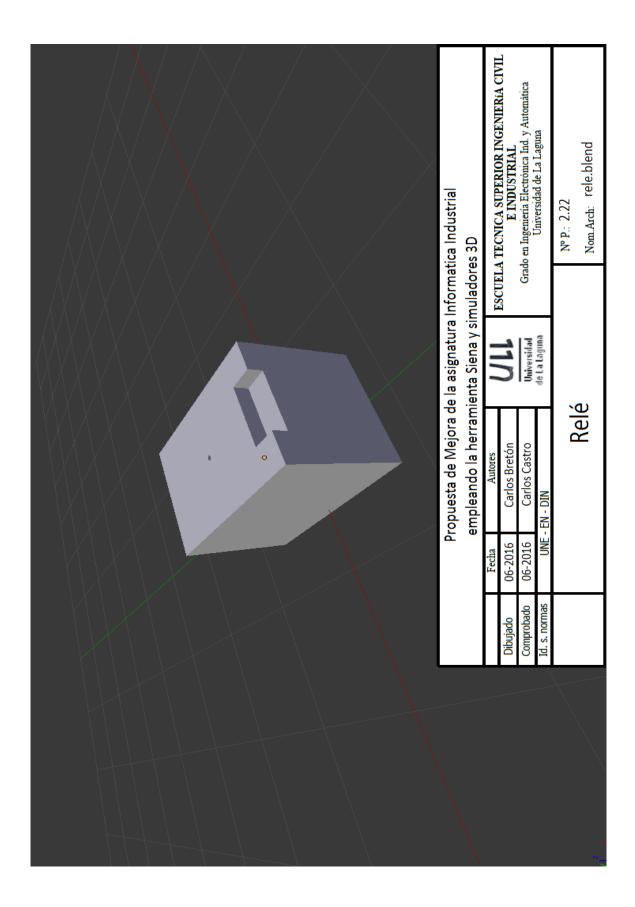


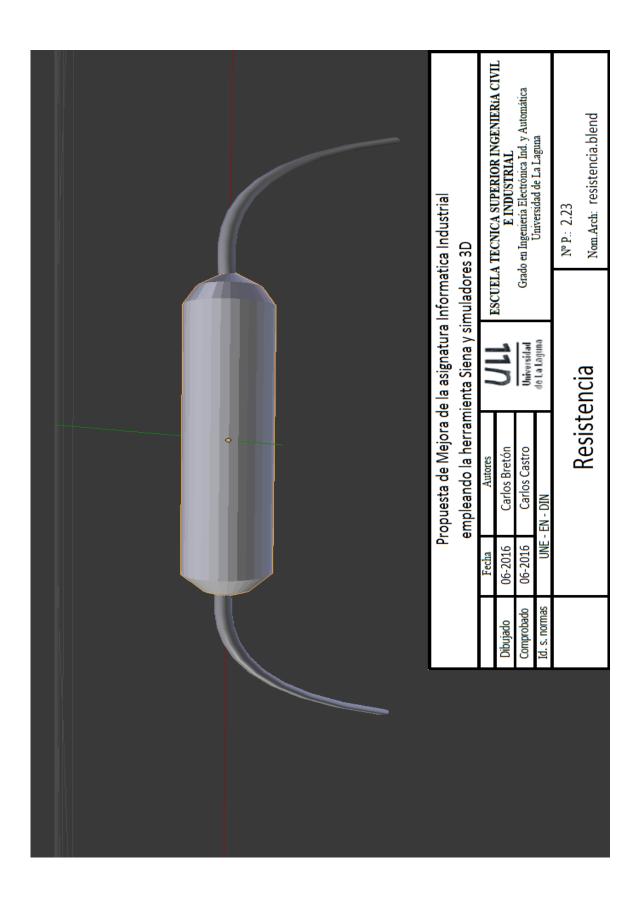


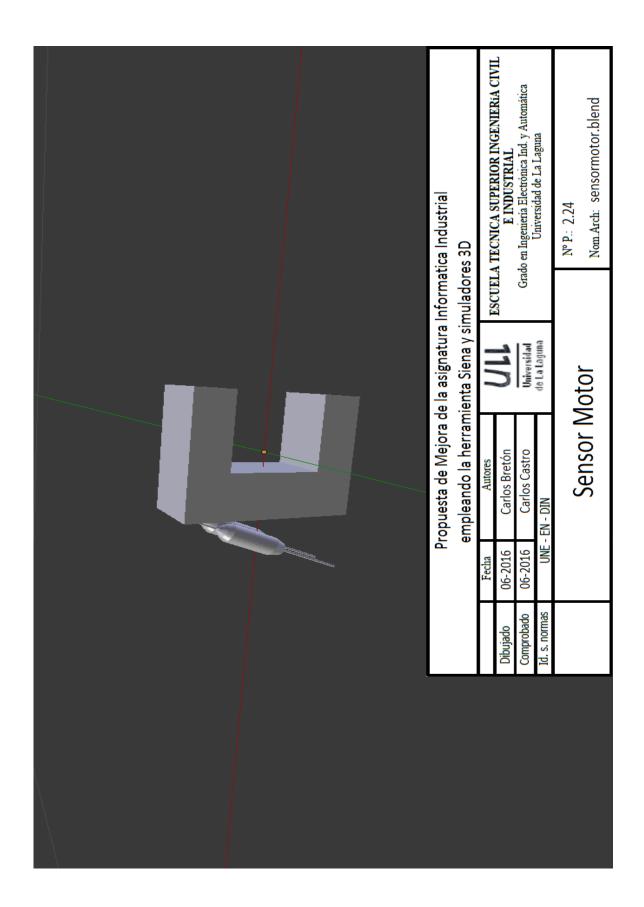


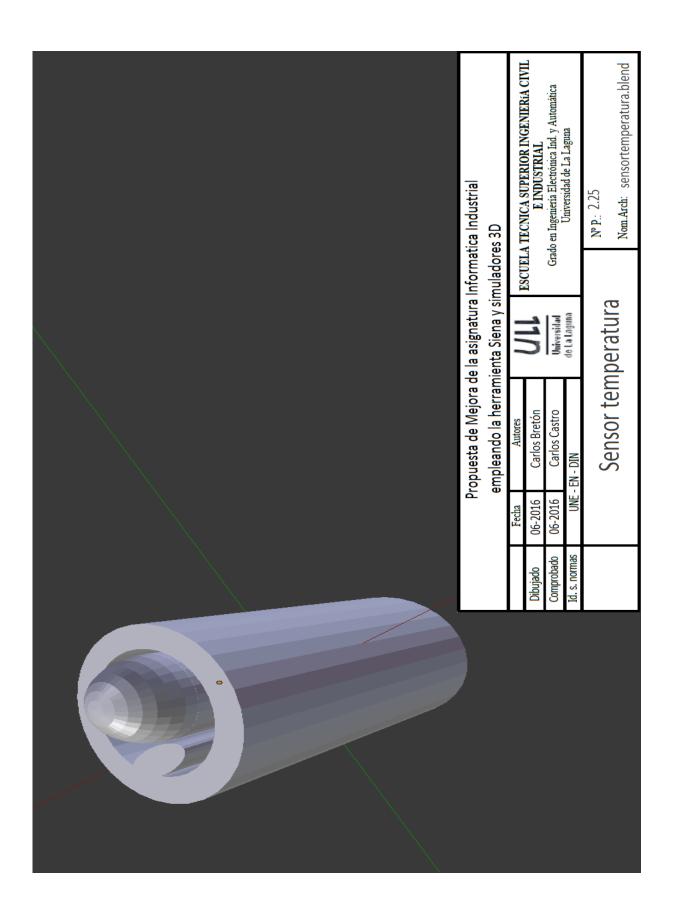


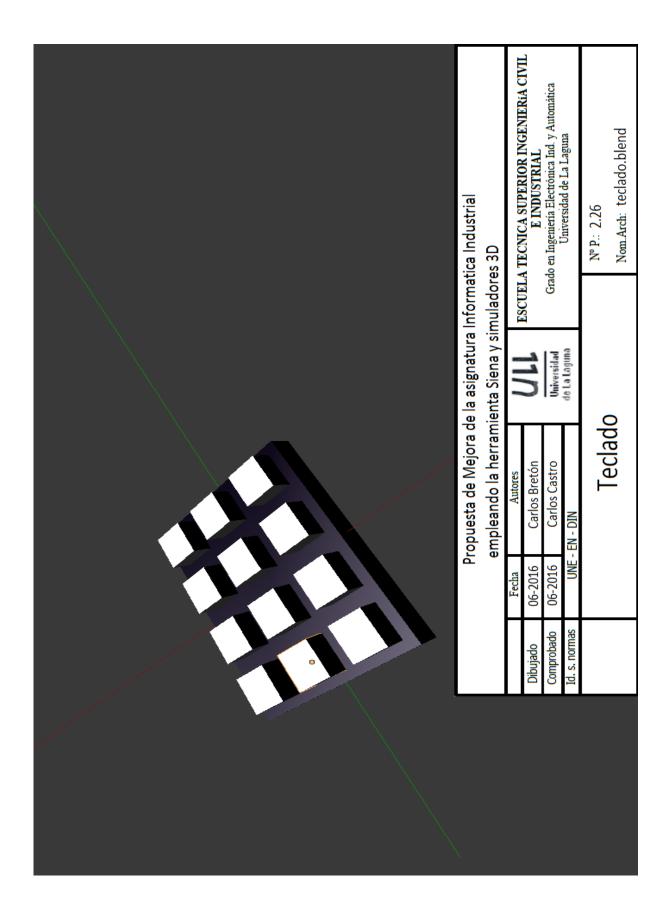


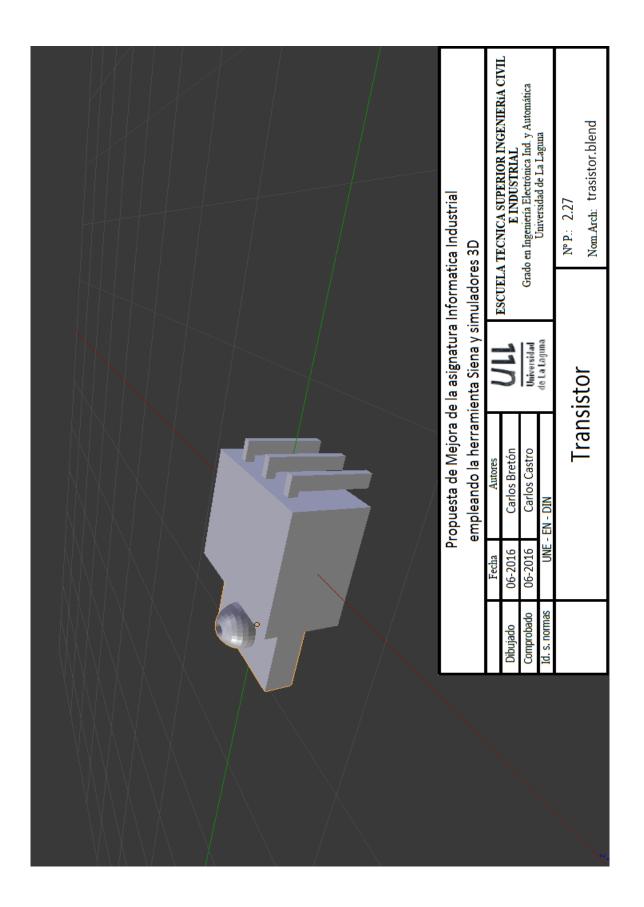


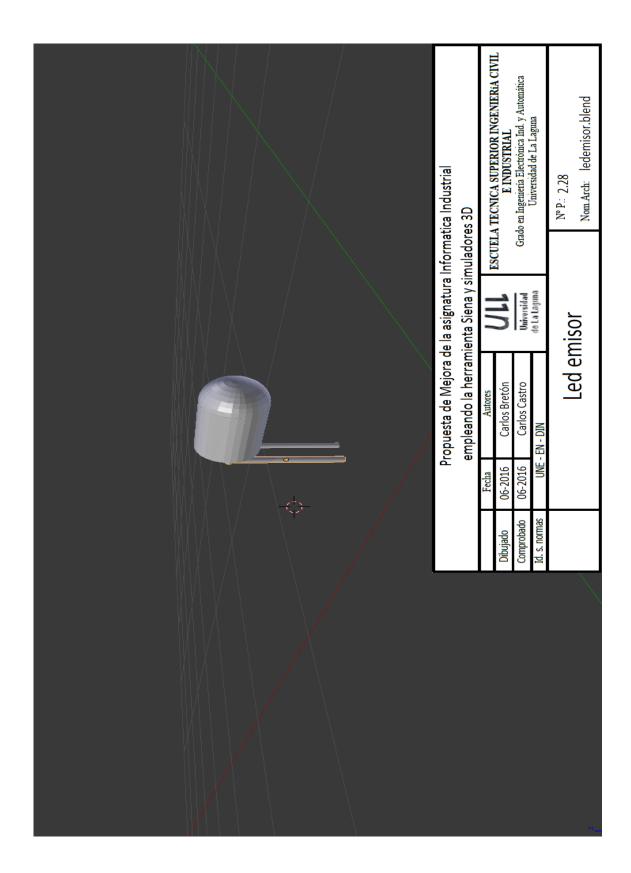














GRADO DE INGENIERÍA ELECTRONICA INDUSTRIAL Y AUTOMÁTICA

TRABAJO FIN DE GRADO

Titulo:

Propuesta de mejora de la asignatura "Informática Industrial" empleando la herramienta SIENA y simuladores 3D

ANEXO II

Pregunta incluidas en el sistema SIENA

Autores:

Carlos Castro Pérez

Carlos Bretón Timón

Tutor:

Evelio González González

Septiembre 2016

Nodo: Introducción a microcontroladores

1º Contenido: Microcontrolador y microporcesador son:

Respuestas:

- Dispositivos relacionados entre sí pero hacen referencia a términos distintos.
- <u>Términos equivalentes.</u>
- Términos que hacen referencia a distintos dispositivos.

2º Contenido: ¿3 módulos esenciales presentes en los microcontroladores?

Respuestas

- Puertos In/Out, conversores A/D y la unidad aritmético lógica.
- Temporizadores, conversores A/D y la unidad aritmético lógica.
- Puertos In/Out, zona de memoria y la unidad aritmético lógica.

3º Contenido: ¿Función principal de un microcontrolador?

Respuestas

- Pisapapeles.
- Controlar.
- Procesar.

4º Contenido: De los tres periféricos, ¿cuál es más importante para un microcontrolador?

Respuestas

- Conversores A/D
- Altavoz.
- Memoria de almacenaje.

5º Contenido: ¿Cuál es la característica principal de un microcontrolador?

- Qué en su zona de memoria tiene el programa a ejecutar en el dispositivo sin dispositivos externos.
- Necesita de un control externo que controle la ejecución adecuada del sistema.
- Es de un tamaño reducido, ideal para todo sistema a pesar de necesitar sistemas externos.

6º Contenido: Fabricante del microcontrolador que se usa en las prácticas.

Respuestas

- Acme.
- Atmel.
- ATmega.

7º Contenido: El concepto de continuidad temporal tiene como objetivo que:

Respuestas

- El microcontrolador sea de larga duración sin necesidad de mantenimiento.
- Se aplica solo a chips no a microcontroladores.
- Que un dispositivo este diseñado para operar de manera ininterrumpida.

8º Contenido: Las prácticas de la asignatura:

Respuestas

- Tienen como objetivo que con unos periféricos controlar al microcontrolador.
- Tienen como objetivo controlar una serie de periféricos a través del ordenador.
- <u>Tienen como objetivo controlar una serie de periféricos a través del</u> microcontrolador.

Nodo: Programación de un microcontrolador y ATmega8515

9º Contenido: Orden ideal del proceso de programación para un microcontrolador.

Respuestas

- Programación, conversión de código fuente a código máquina, depuración y comprobación directa.
- Programación, depuración, conversión de código fuente a código máquina y comprobación directa.
- Programación, depuración, conversión código máquina a código fuente y comprobación directa.

10 Contenido: Qué ventajas tiene programar en el editor de texto.

- Ninguna.
- Se trata de códigos en ficheros de texto plano.
- Es un editor de códigos ideal.

11 Contenido: El software recomendado para usar en la práctica para programar un microcontrolador.

Respuestas

- AVR Studio 4 versión 4.21
- AVR Studio 4 versión 4.20
- AVR Studio 4 versión 4.19

12 Contenido: Otro software gratuito ideal para programar microcontroladores.

Respuestas

- Notepad++
- AVR Studio
- Tavrasm

13º Contenido: Ventajas de AVRISP mkII.

Respuestas

- Conexión por cable coaxial.
- Capaz de programar MC en protoboard.
- Protección contra sobretensiones.

14° Contenido: ¿Qué comprobar si el funcionamiento no es el deseado?

Respuestas

- Conexiones de los cables
- Todas correctas
- Líneas de programación.

15° Contenido: ¿Qué es un lenguaje de programación?

- Es un lenguaje o software diseñado para describir un conjunto de acciones consecutivas que un equipo debe ejecutar.
- Es un compilador estable y potente para la solución de problemas humanos y los entienda físicamente la PC.
- Es un Programa que define un medio de comunicación compartido por un grupo de personas y la PC.

16º Contenido: ¿Cuáles son los tipos de lenguaje de programación?

Respuestas

- Lenguaje Estructural, Máquina, Ensambladores, Fisico-Humano, lenguajes de bajo nivel y de lenguajes de alto nivel
- Únicamente de bajo nivel y lenguajes de alto nivel.
- El lenguaje máquina, Lenguajes ensambladores y de lenguajes de alto nivel

17º Contenido: ¿Qué es un Lenguaje Maquina?

Respuestas

- Es un lenguaje de programación de bajo nivel para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables.
- Es el sistema de códigos directamente interpretable (0 y 1) por un circuito microprogramable, como el microprocesador de una computadora.
- Se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad ejecutora de las máquinas.

18º Contenido: El lenguaje ensamblador es:

Respuestas

- Es un lenguaje de programación de bajo nivel para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables.
- Es el sistema de códigos directamente interpretable (0 y 1) por un circuito microprogramable, como el microprocesador de una computadora.
- Se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad ejecutora de las máquinas.

19° Contenido: El microcontrolador ATmega8515...

Respuestas

- Tiene una capacidad de 16Kb para el programa a ejecutar en binario
- Esta limitada para la mayoría de aplicaciones posibles.
- Solo tiene 8Kb en su memoria.

20° Contenido: El lenguaje ensamblador para microcontroladores:

- Tiene 90 instrucciones.
- Tiene 120 instrucciones.
- Se Accede a sus funciones por medio de registros.

21º Contenido: ¿Qué son los registros?

Respuestas

- Son posiciones en el CAD de almacenamiento.
- Son posiciones de 8bit en la memoria de almacenamiento.
- Su bit más significativo es el 8.

22° Contenido: 0 a 255 valores numéricos o caracteres ASCII, ¿Qué son en el registro?

Respuestas

- La capacidad de almacenamiento.
- La capacidad de los temporizadores.
- La ruta de dirección del microcontrolador.

23º Contenido: Las operaciones de mover datos por la memoria o preparar datos son:

Respuestas

- Parte de los 32 posibles registros de propósito específico del microcontrolador.
- Parte de los 32 posibles registros de propósito general del microcontrolador.
- Ninguna.

24º Contenido: La instrucción "mov":

Respuestas

- Mueve los datos vaciando el espacio en el origen.
- Operación origen, fuente
- Ninguna es correcta

25° Contenido: La instrucción "ser":

Respuestas

- Copia los datos del origen en el destino.
- Limpia los registros en el origen.
- Pone a todos los bits del registro el valor 1.

26° Contenido: Las instrucciones "st" y "ld" son:

- Al igual que "ldi" son operaciones de escritura/lectura.
- Distintas a "ldi" son operaciones de escritura y lectura.
- Ambas son correctas.

27º Contenido: Los puertos de entrada/salida tienen direcciones fijas.

Respuestas

- Son fijos al microcontrolador en AVR
- Los puertos se organizan en 7 bits
- Ninguna es correcta

28° Contenido: Las instrucciones "out" e "in" son operaciones de lectura y escritura respectivamente.

Respuestas

- Verdadero
- Falso
- Y son iguales a la instrucción "ldi"

29° Contenido: La RAM dinámica es una memoria accesible directamente por la CPU.

Respuestas

- Verdadero
- Además, involucra operaciones más rápidas.
- Nada es cierto.

30° Contenido: La instrucción "reall".

Respuestas

- Todas las respuestas son correctas.
- Llama a rutinas con limitados desplazamientos de instrucciones.
- Cambia de orden las subrutinas en el código.

Nodo: Interruptores, led y potenciómetros

31º Contenido: De la práctica, dispositivo DISEN. Si en el módulo de los interruptores mandamos una señal 0001001100 siendo el pin 1 el menos significativo.

- Quiere decir que por los pines 4, 7 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los pines 3, 4 y 7 hay una señal de alta que enciende los leds de los interruptores correspondientes.

• Quiere decir que por los pines 3, 4 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.

32º Contenido: De la práctica, dispositivo DISEN. Si en el módulo de los interruptores mandamos una señal 0011010000 siendo el pin 1 el menos significativo

Respuestas

- Quiere decir que por los pines 3, 4 y 6 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los pines 5, 7 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los pines 4, 6 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.

33º Contenido: De la práctica, dispositivo DISEN. Si en el módulo de los interruptores mandamos una señal 0010010000 siendo el pin 1 el menos significativo

Respuestas

- Quiere decir que por los pines 5 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los pines 3 y 6 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los pines 4 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.

34º Contenido: De la práctica, dispositivo DISEN. Si en el módulo de los leds mandamos una señal 0010100100 siendo el pin 1 el menos significativo

Respuestas

- Quiere decir que los por los leds 3, 5 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que los por los leds 3, 6 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los leds 4, 6 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.

35º Contenido: De la práctica, dispositivo DISEN. Si en el módulo de los leds mandamos una señal 0001001100 siendo el pin 1 el menos significativo

Respuestas

• Quiere decir que por los leds 4, 7 y 8 hay una señal de alta que enciende los leds

de los interruptores correspondientes.

- Quiere decir que los por los leds 3, 4 y 7 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que los por los leds 4, 6 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.

36º Contenido: De la práctica, dispositivo DISEN. Si en el módulo de los leds mandamos una señal 0000110010 siendo el pin 1 el menos significativo

Respuestas

- Quiere decir que por los leds 2, 5 y 6 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los leds 5, 6 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.
- Quiere decir que por los leds 5, 7 y 8 hay una señal de alta que enciende los leds de los interruptores correspondientes.

Nodo: Display de 7 segmentos

37º Contenido: De la práctica, dispositivo DISEN. Si en el módulo de los displays de 7 segmentos mandamos una señal 0000110010 siendo el pin 1 el menos significativo, ¿qué número se muestra?

Respuestas

- <u>Se muestra el número XX22.</u>
- Se muestra el número XX00.
- Se muestra el número 22XX.

38º Contenido: De la práctica, dispositivo DISEN. Si por el módulo de los displays de 7 segmentos mandamos una señal 0010101001 siendo el pin 1 el menos significativo, ¿qué número se muestra?

Respuestas

- Se muestra el número X4X4.
- Se muestra el número 9X9X.
- Se muestra el número X9X9.

39° Contenido: De la práctica, dispositivo DISEN. Si por el módulo de los displays de 7 segmentos mandamos una señal 0011100110 siendo el pin 1 el menos significativo, ¿qué

número se muestra?

Respuestas

- Se muestra el número 9XX9.
- Se muestra el número 666X.
- Se muestra el número X99X.

Nodo: Teclado matricial

40° Contenido: De la práctica, dispositivo DISEN. Si en el módulo del teclado matricial queremos detectar que se ha pulsado la tecla "6", ¿por qué combinación de pines pasara la señal?

Respuestas

- <u>La señal pasará por los pines 5 y 7.</u>
- La señal pasará por los pines 2 y 7.
- La señal pasara por los pines 2 y 5.

41º Contenido: De la práctica, dispositivo DISEN. Si en el módulo del teclado matricial queremos detectar que se ha pulsado la tecla "0", ¿por qué combinación de pines pasara la señal?

Respuestas

- La señal pasará por los pines 4 y 5.
- La señal pasará por los pines 1 y 4.
- La señal pasara por los pines 1 y 5.

42º Contenido: De la práctica, dispositivo DISEN. Si en el módulo del teclado matricial queremos detectar que se ha pulsado la tecla "4", ¿por qué combinación de pines pasara la señal?

- La señal pasará por los pines 1 y 7.
- La señal pasará por los pines 3 y 7.
- La señal pasara por los pines 1 y 3.

Nodo: Motor de corriente continua

43° Contenido: De la práctica, dispositivo DISEN. Si en el módulo del motor DC conectamos un cable mandando una señal de alta por el pin 1, el menos significativo, ¿qué sucede en el dispositivo?

Respuestas

- El motor gira en sentido horario. Los sensores mandan por los pines 4 y 5 una señal cada vez que las aspas pasan a través de ellos.
- El motor gira en sentido anti horario. Los sensores mandan por los pines 4 y 5 una señal cada vez que las aspas pasan a través de ellos.
- El motor gira en sentido horario. Los sensores mandan por los pines 3 y 4 una señal cada vez que las aspas pasan a través de ellos.

44º Contenido: De la práctica, dispositivo DISEN. Si en el módulo del motor DC conectamos un cable mandando una señal de alta por el pin 2, el menos significativo, ¿qué sucede en el dispositivo?

- El motor gira en sentido horario. Los sensores mandan por los pines 4 y 5 una señal cada vez que las aspas pasan a través de ellos.
- El motor gira en sentido anti horario. Los sensores mandan por los pines 4 y 5 una señal cada vez que las aspas pasan a través de ellos.
- El motor gira en sentido anti horario. Los sensores mandan por los pines 3 y 4 una señal cada vez que las aspas pasan a través de ellos.