



**GRADO DE INGENIERÍA ELECTRÓNICA INDUSTRIAL Y  
AUTOMÁTICA**

**TRABAJO DE FIN DE GRADO**

Título:

**Herramienta de apoyo a la enseñanza del estándar IEC 61131 basada  
en la automatización y simulación de la tercera estación de la planta  
Festo y Siena**

**MEMORIA**

Autor/es:

**Jamileth Lotero Nañez**

**Reymar Andrea Vargas Iniesta**

Tutora:

**Carina Soledad González González**

**Marzo 2016**

## Índice

Resumen .....	7
Abstract .....	8
Capítulo 1. Introducción.....	9
Objetivo generales.....	10
Objetivos específicos.....	10
Capítulo 2. Automatización Industrial con PLC´s .....	11
2.1 Introducción .....	11
2.2 Estandarización de los autómatas programables .....	12
2.2.1 Estándar IEC 61131 .....	12
2.2.2 Estándar IEC 61131-3 .....	13
2.3 Entornos de desarrollo para la programación de PLC.....	24
2.3.1 CoDeSys.....	24
2.3.2 STEP7- Microwin .....	38
Capítulo 3. Descripción de la planta FESTO .....	40
3.1 Situación de la planta industrial FESTO .....	40
3.2 Estación de acarreo de piezas.....	41
3.3 Descripción de los elementos de la estación .....	41
3.3.1 Brazo de transferencia.....	41
3.3.2 Almacén de piezas defectuosas .....	43
3.4 Sensores y Actuadores de la planta.....	43
3.4.1 Sensores.....	43
3.4.2 Actuadores.....	44
3.5 Funcionamiento de la estación .....	46
3.6 Conexión con el autómatas.....	50
Capítulo 4. Simulación de estación de acarreo de piezas de la planta Festo.....	51
4.1 Motivo de la simulación.....	51
4.2 Estudio del Software especializado para la simulación .....	51
4.2.1 VariCAd .....	51
4.2.2 SolidWorks.....	53
4.2.3 CATIA.....	54
4.2.4 Inventor .....	55
4.3 Resultado del estudio: .....	56

4.4 Etapas para realizar la simulación.....	57
4.5 Ventajas de la simulación.....	58
Capítulo 5. Inventor.....	59
5.1 ¿Qué es inventor?.....	59
5.2 Características de Inventor.....	59
5.3 Entorno de trabajo.....	61
5.3.1 Interfaz de Inventor.....	61
5.3.2 Entorno de operaciones.....	63
5.3.3 Entorno de ensamblaje.....	67
5.3.4 Entorno de simulación.....	75
5.3.5 Entorno de presentación.....	81
5.3.6 Entorno de documentación.....	87
Capítulo 6. Estudio del Software educativo para el aprendizaje en automatización.....	92
6.1 Introducción.....	92
6.1.1 Objetivo.....	92
6.2 Estudio y elección del programa.....	93
6.2.1 Programación en Python.....	93
6.2.2 Python para la elaboración del programa.....	95
6.2.3 Inconvenientes.....	99
6.2.4 SIENA.....	108
Capítulo 7. Software de aprendizaje para la enseñanza en la ingeniería.....	109
7.1 Objetivo.....	109
7.2 ¿Qué es SIENA?.....	109
7.2.1 ¿Cómo funciona?.....	109
7.2.2 Registrarse en SIENA.....	112
7.3 Elaboración del cuestionario.....	116
7.4 Aplicación de Inventor a SIENA.....	131
Capítulo 8. Conclusiones.....	133
Referencias bibliográficas.....	136
ANEXO I Código de la estación en el lenguaje KOP	
ANEXO II Código de la estación en el lenguaje AWL	
ANEXO III Planos de la estación en Inventor	

## Índice de Figuras y Tablas

Figura 2.1 Estándar Iec 61131-3 .....	13
Figura 2.2 Ejemplo De Un Diagrama De Contactos.....	16
Figura 2.3 Líneas De Alimentación De Un Diagrama Ld.....	16
Figura 2.4 Distribución De Elementos En El Diagrama De Contactos .....	17
Figura 2. 5 Ejemplo De Un Diagrama De Bloques Funcionales .....	18
Figura 2.6 Tipos De Bloques A) Bloque De Función B) Función.....	19
Figura 2.7 Ejemplo De Una Lista De Instrucciones.....	20
Figura 2.8 Ejemplo De Un Texto Estructurado.....	22
Figura 2.9 Interfaz De Codesys.....	25
Figura 2.10 Ventana De Configuraciones Del Sistema De Destino .....	25
Figura 2 11 Ventana De Nuevo Módulo .....	26
Figura 2.12 Entorno De Trabajo .....	27
Figura 2.13 Navegador De Pou's .....	27
Figura 2.14 Opción Recursos .....	28
Figura 2.15 Panel De Herramientas .....	29
Figura 2.16 Declaración De Variables Locales.....	29
Figura 2.17 Sintaxis Para Declarar Una Variable .....	30
Figura 2.18 Declaración De Las Variables Globales .....	30
Figura 2.19 Editor Del Programa .....	31
Figura 2.20 Menú De Visualizaciones .....	32
Figura 2.21 Ventana Nueva De Visualización .....	32
Figura 2.22 Interfaz Scada .....	33
Figura 2.23 Entorno De Trabajo Scada.....	33
Figura 2.24 Configurar Elemento.....	34
Figura 2.25 Configurar Nombre Del Elemento.....	34
Figura 2.26 Configurar Color Del Elemento.....	35
Figura 2.27 Asignación De La Variable Al Elemento .....	35
Figura 2.28 Resultado Del Ejemplo .....	36
Figura 2.29 Opción En Línea .....	36
Figura 2.30 Opciones Inicio De Sesión Y Estado De Los Elementos .....	37
Figura 2.31 Comprobación Del Funcionamiento.....	37
Figura 2.32 Interfaz De Step 7 Microwin .....	38
Figura 3.1 Estaciones De La Planta Festo.....	40
Figura 3.2 Subsistemas De La Estación .....	41
Figura 3.3 Sistema De Rotación.....	42
Figura 3.4 Elemento Horizontal Y Portador .....	43
Figura 3.5 Sensores Del Brazo Giratorio .....	43
Figura 3.6 Sensores Del Elemento Horizontal .....	44
Figura 3. 7 Sensores Del Elemento Vertical .....	44
Figura 3.8 Válvula Que Activa Al Elemento Vertical .....	45
Figura 3.9 Brazo En Posición Inicial .....	46
Figura 3.10 Elemento Horizontal Extendido .....	46
Figura 3.11 Pieza En Ventosa .....	47
Figura 3.12 Brazo En Posición Posterior .....	47

Figura 3.13 Pieza No Defectuosa En Estación 4.....	48
Figura 3.14 Retracción De Elementos Vertical Y Horizontal.....	48
Figura 3.15 Pieza Defectuosa En Almacén.....	48
Figura 3.16 Luz De Parada De Emergencia.....	49
Figura 3.17 Autómata S7-200.....	50
Figura 4. 1 Interfaz Varicad.....	53
Figura 4.2 Interfaz Solidworks.....	54
Figura 4. 3 Interfaz Catia.....	55
Figura 4.4 Interfaz Inventor.....	56
Figura 5. 1 Interfaz Inventor.....	62
Figura 5 2 Interfaz De Trabajo Inventor.....	62
Figura 5.3 Ventosa.....	63
Figura 5.4 Formato De Archivo.....	64
Figura 5.5 Boceto De La Ventosa.....	64
Figura 5.6 Desfase De Planos.....	65
Figura 5.7 Solevación De Pieza.....	65
Figura 5.8 Extrusión De La Pieza.....	66
Figura 5.9 Empalme De La Pieza.....	66
Figura 5. 10 Pieza Terminada.....	67
Figura 5. 11 Restricciones De Ensamble.....	67
Figura 5.12 Análisis De Restricciones.....	68
Figura 5.13 Restricción Redundante.....	69
Figura 5.14 Error En La Restricción.....	69
Figura 5.15 Elección De Conjunto Soldado.....	70
Figura 5.16 Insertar Piezas.....	70
Figura 5.17 Elección De Piezas.....	71
Figura 5.18 Elemento Fijo.....	71
Figura 5. 19 Restricción De Coincidencia.....	72
Figura 5.20 Armadura Sobre Mesa.....	72
Figura 5. 21 Segunda Restricción.....	73
Figura 5.22 Armadura En Posición Correcta.....	73
Figura 5.23 Restricción Inserción.....	74
Figura 5.24 Ensamble Completo.....	74
Figura 5.25 Acceso A Inventor Studio.....	75
Figura 5.26 Herramientas Inventor Studio.....	75
Figura 5.27 Creación De Nueva Animación.....	76
Figura 5.28 Animar Restricción.....	77
Figura 5.29 Ventana Para Configurar Animación De Restricción.....	77
Figura 5.30 Creación De Cámara.....	78
Figura 5.31 Selección De Cámara.....	78
Figura 5.32 Configuración De Cámara.....	79
Figura 5.33 Producir Vídeo.....	79
Figura 5.34 Renderizar Animación.....	80
Figura 5.35 Video Renderizado Estación De Acarreo De Piezas.....	80
Figura 5.36 Interfaz De Autodesk Inventor Publisher.....	82
Figura 5. 37 Nuevo Proyecto.....	82

Figura 5.38 Elección De Archivo.....	83
Figura 5.39 Grupo De Instantáneas.....	83
Figura 5.40 Opción Nueva Instantánea.....	83
Figura 5.41 Opción Explosión.....	84
Figura 5.42 Despiece De Elemento.....	84
Figura 5.43 Restaurar Ensamblaje.....	85
Figura 5.44 Reproducir Presentación.....	85
Figura 5.45 Modificar Tiempo De Transición.....	86
Figura 5.46 Activar Cámara.....	86
Figura 5.47 Formatos De La Presentación.....	87
Figura 5.48 Opción Dibujo.....	88
Figura 5.49 Plantilla Predeterminada.....	88
Figura 5.50 Editar Cajetín.....	89
Figura 5.51 Realizar Nuevo Cajetín.....	89
Figura 5.52 Obtener Plano De Una Pieza.....	90
Figura 5.53 Configurar Plano.....	90
Figura 5.54 Creación De Vistas.....	91
Figura 5.55 Opciones Para Los Planos.....	91
Figura 6.1 Programación En C.....	94
Figura 6.2 Programación En C++.....	95
Figura 6.3 Programación En Python.....	95
Figura 6.4 Editor De Texto Geany.....	96
Figura 6.5 Pregunta Elaborada En Python.....	97
Figura 6.6 Consola De Windows.....	97
Figura 6.7 Ejecución Del Programa.....	98
Figura 6.8 Pregunta En Python.....	98
Figura 6.9 Respuesta Incorrecta.....	99
Figura 6.10 Respuesta Correcta.....	99
Figura 6.11 Ejemplo De Tkinter.....	100
Figura 6.12 Ventana Creada Desde Tkinter.....	101
Figura 6.13 Primer Juego De Pygame.....	102
Figura 6.14 Entorno De Siena.....	108
Figura 7.1 Siena Social.....	113
Figura 7.2 Acceso Usuarios.....	113
Figura 7.3 Nuevo Usuario.....	114
Figura 7.4 Profesor En Siena.....	115
Figura 7.5 Menú Para Los Profesores.....	115
Figura 7.6 Menú Para Los Alumnos.....	116
Figura 7.7 Interfaz De Compendiumld.....	117
Figura 7.8 Nuevo Proyecto.....	118
Figura 7.9 Acceso Al Proyecto.....	118
Figura 7.10 Creación De Nodos.....	119
Figura 7.11 Mapa Conceptual Para Siena.....	120
Figura 7.12 Exportar A Xml.....	120
Figura 7.13 Lista De Asignaturas.....	121
Figura 7.14 Menú De La Asignatura.....	121

Figura 7.15 Importar Mapa A Siena.....	122
Figura 7. 16 Generar Imagen Del Mapa.....	122
Figura 7.17 Mapa De La Asignatura.....	123
Figura 7.18 Elaboración De La Pregunta.....	124
Figura 7.19 Pregunta Creada En Siena.....	125
Figura 7. 20 Nodos Disponibles Para La Asignatura.....	126
Figura 7.21 Nodos Relacionados.....	127
Figura 7.22 Lista De Preguntas.....	127
Figura 7.23 Preguntas Elaboradas Para El Cuestionario.....	128
Figura 7. 24 Interfaz De Siena Para Alumnos.....	128
Figura 7.25 Lista De Trabajos De La Asignatura.....	129
Figura 7.26 Lista De Grupos.....	129
Figura 7.27 Lista De Asignaturas.....	130
Figura 7.28 Pregunta Del Cuestionario.....	130
Figura 7.29 Canal De Youtube.....	131
Figura 7.30 Preguntas Con Enlace A Youtube.....	132
Figura 7.31 Vídeo Del Estado Inicial De La Estación.....	132
Tabla 2.1 Elementos Del Diagrama De Contactos.....	17
Tabla 2.2 Tabla De Operadores Il. Tomado De [5].....	21
Tabla 2.3 Operadores De Lenguaje St. Tomado De [6].....	23
Tabla 2.4 Lista De Instrucciones En El Lenguaje St. Tomado De [6].....	23
Tabla 3.1 Entradas Y Sensores Del Autómata.....	50
Tabla 3.2 Salidas Y Actuadores Del Autómata.....	50

## Resumen

Para una correcta y efectiva enseñanza en el campo de la ingeniería, es necesario que los alumnos posean herramientas que les ayude a profundizar los conocimientos teóricos y prácticos aprendidos a lo largo de su carrera profesional.

En este proyecto se expone una alternativa de aprendizaje para la enseñanza de los estudiantes del último curso del Grado de Ingeniería Electrónica Industrial y Automática, que cursen la asignatura de automatización industrial avanzada en la universidad de la Laguna.

Mediante la aplicación de un software educativo el estudiante reforzará los conocimientos referentes al temario de dicha asignatura: la normativa internacional IEC 61131 que define y regula la programación de los autómatas programables, entornos de programación actualmente utilizados en el mercado industrial: Step 7 Microwin y CoDeSys, además de los lenguajes de programación definidos por la norma.

Como herramienta complementaria para este método de aprendizaje, se utilizará un software de modelado en 3D, con el que se elaborará la simulación de la tercera estación de la planta Festo, contenido práctico que hace parte del temario de la asignatura de automatización industrial. Con este programa se creará contenido multimedia, videos que mostrarán el funcionamiento correcto de la estación de acarreo de piezas, lo que contribuirá a una preparación mejor de los alumnos antes de realizar las prácticas de la asignatura en cuestión.



## Abstract

For a proper and effective training in the field of engineering, it is necessary for students to possess certain tools that will help them broaden the theoretical and practical knowledge they acquired throughout the course of their professional career.

In this project we expose a learning alternative for the education of students in the last grade of the Industrial Electronic Engineering and Automation degree that are currently taking on the subject “Advanced Industrial Automation” in the University of La Laguna.

Using an educational software the students will reinforce their knowledge related to the syllabus of the aforementioned subject: the IEC 61131 international standards that define and regulate the programming of programmable automatons, programming environments currently used in the industrial market: Step 7 Microwin and CoDeSys, as well as the programming languages defined by the standards.

As a complementary tool for this learning method, we will use a 3D modeling software, with which we will carry out the simulation of the third station of the Festo plant, a practical content that is part of the syllabus of the Industrial Automation subject. With this program we will create multimedia content, videos that will show the proper working of the piece carriage station, which will contribute to a better preparation of students before they perform the practices of the mentioned subject.

## Capítulo 1. Introducción

Hoy en día las metodologías empleadas en el mundo de la enseñanza se encuentran en constantes cambios. Éstas ya no se basan solamente en clases teóricas donde el alumno se ve “obligado” a escuchar a un profesor, aprenderse lo que han dado en clase y luego olvidarlo como suele pasarle a la mayoría de los alumnos, desde que nos encontramos en el colegio hasta la universidad.

El Espacio Europeo de Educación Superior (EEES) presenta diferentes metodologías necesarias para mejorar la calidad de la enseñanza, que se pueden aplicar en la ingeniería [1]: clases teóricas, prácticas en el laboratorio, aprendizaje cooperativo, aprendizaje basado en problemas (ABP), seminarios, tutorías y Tecnologías de la Información y la Comunicación (TICs).

Éste introdujo cambios en los métodos de la docencia universitaria para mejorar el proceso de enseñanza-aprendizaje. Estos cambios impulsaron el desarrollo de materiales didácticos digitales como apoyo en la docencia con el fin de favorecer el autoaprendizaje activo de los estudiantes.

La implementación de las TIC y la utilización de los servicios y herramientas de la Web 2.0 en las universidades, permite elaborar materiales, contenidos y otras formas de relacionarse con otras personas a distancia.

Debido a la evolución de las nuevas tecnologías, las universidades han ido cambiando la metodología de la enseñanza con el fin de mejorar la calidad de la misma. En la actualidad un alumno universitario dispone de ordenadores portátiles y de acceso a internet tanto en la misma universidad como en su hogar, esto permite que él pueda prepararse una asignatura de forma autónoma si disponen de los medios necesarios para poder hacerlo (materiales, actividades, etc.). Esto puede ayudar a aquellos estudiantes que no puedan acudir a las clases teóricas.

En ingeniería, la realización de prácticas en el laboratorio es el pilar más importante para los alumnos, ya que es aquí donde éstos pueden aplicar los conocimientos adquiridos en las clases teóricas. Sin embargo, los laboratorios no siempre disponen de los puestos necesarios para que todos los alumnos de una clase puedan realizar sus prácticas, lo que termina convirtiéndose en una práctica grupal que si bien puede beneficiarlos a la hora de

intercambiar ideas, puede perjudicarlos ya que en los trabajos grupales lo normal es que no todos los integrantes participen en la realización de la práctica de la misma forma.

A través de la simulación se pueden plantear modelos que se acerquen a la realidad de lo que se trabaja en el laboratorio, esto permite al alumnado recurrir a la previa simulación para afianzar sus conocimientos e incluso completar la misma práctica en dicho entorno simulado.

## **Objetivo generales**

El presente proyecto tiene como objetivo principal la creación de un método de aprendizaje que contribuya a la enseñanza en la ingeniería, concretamente a los alumnos de último curso del Grado De Ingeniería Electrónica Industrial y Automática.

Se empleará una herramienta creada específicamente para el autoaprendizaje y la autoevaluación, con la que el estudiante afianzará los temas teóricos y prácticos de la asignatura de Automatización Industrial Avanzada.

Como herramienta complementaria para realizar este método, se utilizará un software de modelado en 3D. Con éste se realizará la simulación del funcionamiento de la tercera estación de la planta industrial Festo, objeto de estudio práctico de la asignatura en cuestión.

## **Objetivos específicos**

- Elaboración del código de automatización de la tercera estación de la planta industrial Festo en los lenguajes de programación KOP y AWL, y verificación del funcionamiento correcto del código en la planta real ubicada en el laboratorio de automatización de la universidad de la Laguna.
- Estudio y elección del software de modelado en 3D, para realizar la simulación del funcionamiento de la tercera estación de la planta Festo y material multimedia complementario para el método de aprendizaje de la enseñanza en la ingeniería

Los alumnos podrán observar a través de la simulación realizada, las distintas situaciones que pueden llevarse a cabo en la estación como el funcionamiento

correcto de la misma y algunos de los fallos que puedan presentarse.

- Elaboración de un cuestionario enfocado a los temas impartidos en la asignatura Automatización Industrial Avanzada. Las preguntas desarrolladas se introducirán en el software educativo utilizado, como método de aprendizaje de la enseñanza en la ingeniería.

Los temas a tratar serán: el Estándar Internacional que regula los autómatas programables ( IEC 61131), los entornos de programación Step 7 Microwin y CoDeSys, los cuatro lenguajes de programación regulados para la programación de autómatas y la automatización de la tercera estación de la planta Festo.

- Elección del software educativo. Se estudiará la posibilidad de realizarlo con un lenguaje de programación o utilizar directamente uno ya elaborado.

## Capítulo 2. Automatización Industrial con PLC's

### 2.1 Introducción

Con la aparición de los controladores lógicos programables se logró grandes avances en la automatización industrial. Estos dispositivos resultaron ser la salida para desplazar el difícil y costoso método de automatización que existía en los años 60 “la lógica cableada”.

El principal fallo que presentaba este método era ser un sistema basado en relés. Los cuales tenían un tiempo de vida limitado por lo que el mantenimiento tenía que ser frecuente y estricto. Además, el problema aumentaba cuando el proceso a automatizar era más grande ya que la detección de errores era más tediosa y lenta debido al uso de tantos relés.

Los primeros PLC's producidos fueron en el año 1968 por la empresa Bedford Associates. Se les dio el nombre MODICON (*MODular DIgital CONtroller*), programados en lenguaje Ladder, lenguaje que guardaba mucha similitud con el diagrama esquemático de la lógica de relés. El objetivo era fabricar PLC's que fueran fácilmente programables, resistentes a ambientes difíciles y con una vida útil larga. Así se obtuvo un método más eficaz para la automatización de procesos, “la lógica programada”, en donde si un proceso debía de modificarse en algún punto, se cambiaba el programa que controlaba dicho punto específico de manera sencilla, debido a la facilidad del dialogo entre la máquina y el operario.

A partir de los años 80, surge la necesidad de estandarizar los autómatas programables. El primer intento fue El Protocolo de fabricación automatizada MAP (*Manufacturing Automation Protocol*), una especificación impulsada por General Motors con el fin de obtener una arquitectura de comunicaciones independiente del fabricante. En los años 90 se introdujeron nuevos protocolos y se mejoraron algunos anteriores. Actualmente se cuenta con el estándar IEC 61131, descrito como un gran avance en la estandarización de los autómatas programables y sus periféricos.

## **2.2 Estandarización de los autómatas programables**

### **2.2.1 Estándar IEC 61131**

Estándar internacional publicado por la Comisión Electrotécnica Internacional. Es un conjunto de normas e informes técnicos referido a los PLC's y sus periféricos asociados, tales como herramientas de programación y depuración, elementos de interface Hombre-Máquina.

Este documento define las características principales para la selección y aplicación de los PLC's, especifica los requisitos mínimos para las características funcionales, condiciones de servicio, aspectos constructivos, aspectos de seguridad propios de los autómatas como dispositivo físico, detalla los ensayos aplicables a los PLC's y sus periféricos, además de definir los lenguajes de programación que se deben de utilizar para estos.

La norma IEC 61131 se divide en ocho partes:

Parte1: Información General

Parte 2: Especificaciones y ensayos de los equipos

Parte 3: Lenguajes de Programación

Parte 4: Guías de Usuario.

Parte 5: Comunicaciones

Parte 7: Programación en lógica difusa.

Parte 8: Guías para aplicación e implementación de lenguajes de programación.

### 2.2.2 Estándar IEC 61131-3

En este capítulo se hablará de dos entornos de programación para autómatas programables muy importantes, Step 7 Microwin y CoDeSys. Por lo que es de suma importancia conocer que tipos lenguajes de programación están homologados por la Normativa Internacional. La parte 3 del estándar IEC 61131, es donde se definen dichos lenguajes, a continuación se detalla su contenido.

Normativa que define y regula la programación de PLC's, específica los lenguajes de programación de uso estándar para los autómatas, define las reglas de sintaxis y semántica y las instrucciones fundamentales de cada lenguaje.

Este estándar se puede dividir en dos temas, el primero habla acerca de los elementos comunes de la programación en controladores programables y el segundo tema trata los diferentes lenguajes de programación establecidos por la norma.

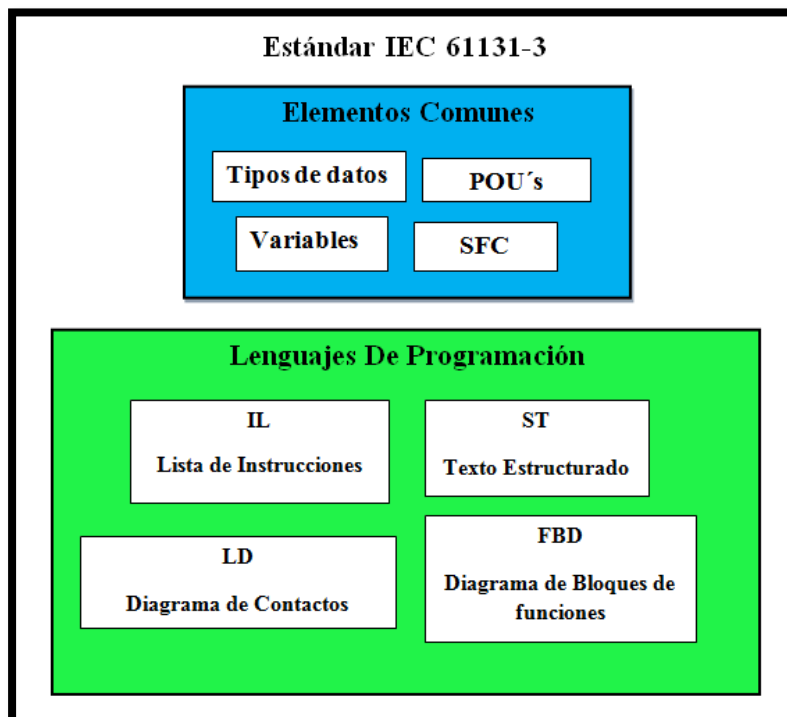


Figura 2.1 Estándar IEC 61131-3

#### Elementos comunes

El estándar establece un conjunto de conceptos globales para todos sus lenguajes de programación, a continuación se detalla cada uno:

## Tipos de datos

Definir la tipología del parámetro usado es muy importante de cara a la programación, se evita errores como realizar operaciones matemáticas con variables de unidades diferentes.

Existen dos tipos de datos:

- Los tipos de datos comunes: booleanos (BOOL), enteros (INT), reales (REAL), byte (BYTE), palabra (WORD), cadenas de caracteres (STRING), fecha (DATE), hora \_ del \_ día (TIME\_OF\_DAY).
- Los tipos de datos derivados: son definidos por el usuario basándose en los tipos de datos comunes.

## Variables

Las variables equivalen a datos que cambian constantemente, se pueden declarar como tipos de datos comunes o como tipos de datos derivados. Las variables se encuentran limitadas a la unidad de organización en la cual son declaradas. Esto significa que sus nombres pueden ser reutilizados en otras partes sin conflictos, eliminando una frecuente fuente de errores.

## Unidades de Organización de Programa (POU's)

Dentro de la norma IEC 61131-3, el conjunto de programas, bloques de Funciones y funciones se designa como POU's

- **Funciones**

IEC define Funciones estándar y funciones definidas por el usuario. Las funciones estándar son por ejemplo: ADD (suma), ABS (absoluto), SQRT (cuadrado) SIN (seno), etc. Las funciones definidas por el usuario (basadas en las funciones estándar), una vez definidas ser usadas indefinidamente en cualquier POU.

- **Bloques de Funciones**

Representan funciones de control especializadas, contienen datos e instrucciones y a diferencia de las funciones pueden guardar los valores de las variables. Tienen una interfaz de entradas y salidas definida y un código interno oculto, tal como un circuito integrado. Un ejemplo de estos son los contadores y temporizadores.

- **Programas**

Son la agrupación de elementos y construcciones de un lenguaje de programación, realizado para una tarea determinada. Un programa puede contener, aparte de la declaración de tipos de datos, variables y su código interno, distintas instancias de funciones y bloques funcionales.

### **Gráfico Funcional Secuencial (SFC)**

Describen gráficamente el comportamiento secuencial de un programa de control. SFC ayuda a estructurar la organización interna de un programa y a descomponer un problema en partes manejables, manteniendo simultáneamente una visión global. Los elementos del SFC proporcionan un medio para subdividir una POU de un autómata programable en un conjunto de etapas y transiciones interconectadas por medio de enlaces directos. Cada etapa lleva asociados un conjunto bloques de acción y a cada transición va asociada una condición de transición que cuando se cumple, causa la desactivación de la etapa anterior a la transición y la activación de la siguiente. Los bloques de acción permiten realizar el control del proceso. Cada elemento puede ser programado en alguno de los lenguajes IEC, incluyéndose el propio SFC. Dado que los elementos del SFC requieren almacenar información, las únicas POU's que se pueden estructurar utilizando estos elementos son los bloques funcionales y los programas. [4]

### **Lenguajes de programación**

El estándar IEC 61131-3 define la sintaxis y semántica de cuatro lenguajes de programación, dos son de tipo textual y dos de tipo gráfico. Estos nos permiten realizar cualquier programa sobre un PLC, incluso se puede trabajar sobre distintos lenguajes pasando de un lenguaje a otro sin ningún inconveniente, debido a que los cuatro lenguajes están interrelacionados. La elección del lenguaje de programación dependerá del problema a tratar, los conocimientos del programador, de la estructura del sistema de control y del grado de comunicación con otros departamentos de la empresa

El motivo por el que se definen estos lenguajes es para evitar particularidades por parte de las empresas que fabrican PLC's.

Seguidamente se explicará cada uno de los lenguajes de programación:



**Diagrama de contactos (*Ladder Diagram, LD*)**

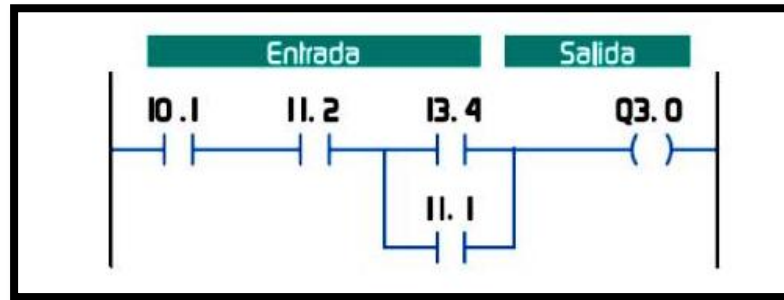


Figura 2.2 Ejemplo de un Diagrama de contactos

Es un lenguaje de origen estadounidense, se basa en símbolos gráficos ubicados en redes (*networks*), comúnmente se le conoce como diagrama "escalera" porque se asemeja a esta, con dos rieles verticales (de alimentación) y "escalones" (líneas horizontales), en las que hay circuitos de control que definen la lógica a través de funciones. Empezando por la izquierda se va realizando una consulta al estado de las entradas y se combina lógicamente esa entrada con el estado anterior.

Toda red en LD está compuesta por dos líneas de alimentación. La línea de inicio (barra izquierda) y la línea final (barra derecha). Estos carriles de alimentación son las líneas de suministro de energía L1 y L2 para los circuitos de corriente alterna o un valor determinado de voltaje y tierra para los circuitos de CC.

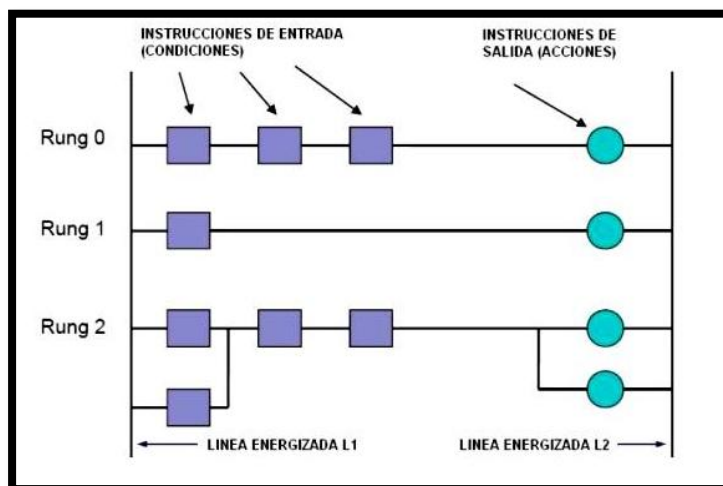


Figura 2.3 Líneas de alimentación de un diagrama LD

- **Elementos de programación**

Para programación mediante este lenguaje es necesario conocer cada uno de sus elementos. En la siguiente tabla podemos observar los símbolos de los elementos básicos junto con sus respectivas descripciones.

Símbolo	Nombre	Descripción
	Contacto NA	Se activa cuando hay un uno lógico en el elemento que representa, esto es, una entrada (para captar información del proceso a controlar), una variable interna o un bit de sistema.
	Contacto NC	Su función es similar al contacto NA anterior, pero en este caso se activa cuando hay un cero lógico, cosa que deberá de tenerse muy en cuenta a la hora de su utilización.
	Bobina NA	Se activa cuando la combinación que hay a su entrada (izquierda) da un uno lógico. Su activación equivale a decir que tiene un uno lógico. Suele representar elementos de salida, aunque a veces puede hacer el papel de variable interna.
	Bobina NC	Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico. Su activación equivale a decir que tiene un cero lógico. Su comportamiento es complementario al de la bobina NA.
	Bobina SET	Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina en RESET. Sirve para memorizar bits y usada junto con la bina RESET dan una enorme potencia en la programación.
	Bobina SET	Permite desactivar una bobina SET previamente activada.

Tabla 2.1 Elementos del Diagrama de contactos

- **Programación**

El esquema de la Figura 2.4 representa la estructura general de la distribución de todo programa hecho en este lenguaje, contactos a la izquierda y bobinas a la derecha.

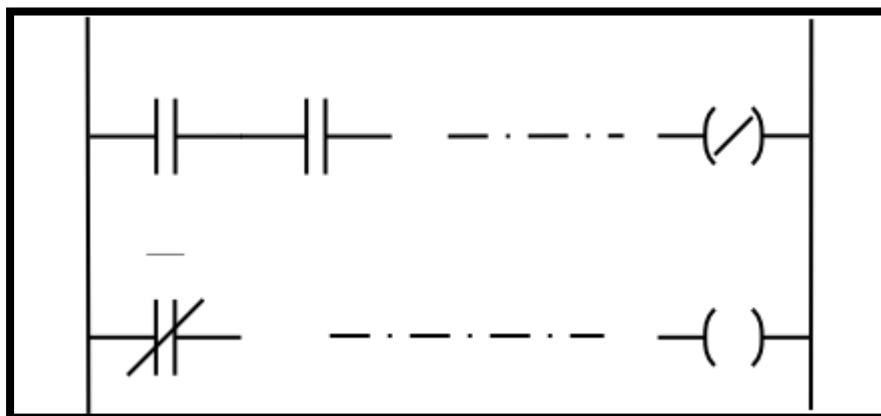


Figura 2.4 Distribución de elementos en el diagrama de contactos

Las operaciones lógicas en este lenguaje son muy parecidas a la representación de un esquema eléctrico, donde se alimenta una bobina que se encuentra al final del circuito si se halla un camino para el paso de corriente desde la parte superior hasta la bobina

El orden de ejecución es generalmente de arriba hacia abajo y de izquierda a derecha, primero los contactos y luego las bobinas, de manera que al llegar a éstas ya se conoce el valor de los contactos y se activan si procede. El orden de ejecución puede variar de un controlador a otro, pero siempre se respetará el orden de introducción del programa, de manera que se ejecuta primero lo que primero se introduce.

### Diagramas de Bloques Funcionales (*Function Block Diagram, FBD*)

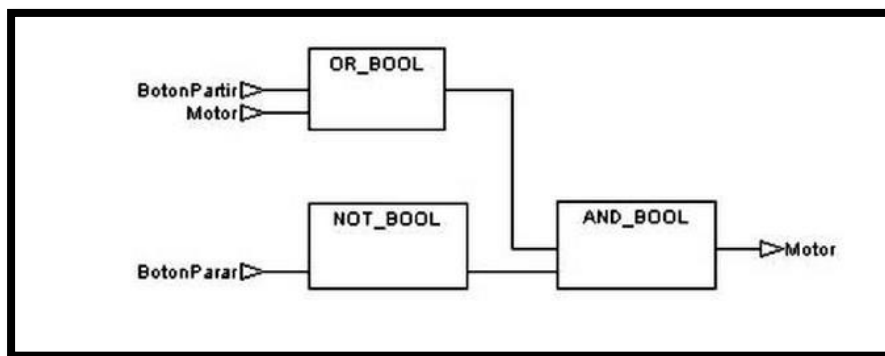


Figura 2. 5 Ejemplo de un Diagrama de Bloques Funcionales

Es un lenguaje gráfico que se utiliza para programar procedimientos complejos mediante la unión de bloques funcionales prediseñados, en este lenguaje es de suma importancia además de los datos booleanos, los datos enteros y de coma flotante.

Este lenguaje describe una función entre las variables de entrada y variables de salida, como un conjunto de bloques elementales, que están conectados por líneas de conexión, de forma análoga a la conexión rieles del lenguaje LD, con la diferencia de que las salidas lógicas no requieren incorporar una bobina de salida, ya que la salida es representada por una variable asignada a la salida del bloque.

El estándar IEC 61131-3 permite al usuario construir sus propios bloques de funciones, de acuerdo a los requerimientos del programa de control.

Es ampliamente utilizado en la industria de procesos, especialmente en aplicaciones que implican flujo de información o datos entre componentes de control.

- **Elementos de programación**

### Bloques

Se usan para llamar a otras POU's de tipo bloque funcional o funciones, se representan mediante cajas rectangulares. Los bloques funcionales pueden tener varias entradas y varias salidas, deben tener un nombre de instancia en la parte superior y el nombre del objeto del cual es instancia dentro de la caja. Las funciones pueden tener múltiples entradas pero solo una salida (valor de retorno), no poseen nombre de instancia ya que son llamadas directamente para que devuelvan un valor. Como se muestra en la Figura 2.6 (a) una instancia del bloque funcional CTD llamada Nombre\_instancia, se puede observar que posee 3 entradas CD, LD y PV y dos salidas Q y CV. En la Figura 2.6 (b) aparece una función llamada SR, tiene 2 entradas S1 y R, y una salida Q. [3.1.1]

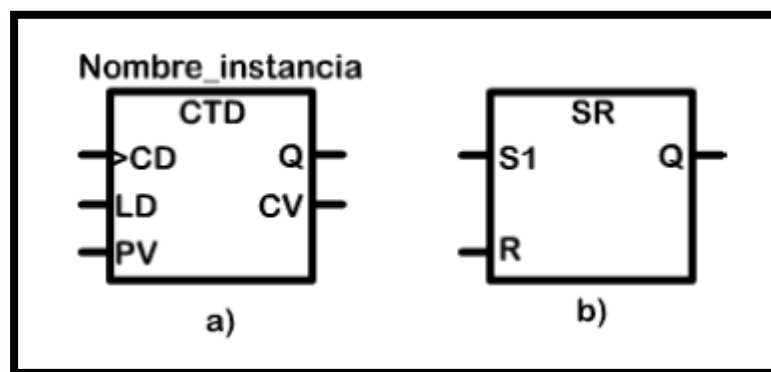


Figura 2.6 Tipos de Bloques a) Bloque de función b) Función

- **Programación**

El procedimiento para la evaluación del código se realiza de forma similar al lenguaje LD. Cada red se evalúa de izquierda a derecha continuando con las redes de abajo. Para realizar la evaluación de redes se debe tener en cuenta estos tres pasos:

1. Evaluar todas las entradas de los elementos de la red antes de ejecutar el elemento.
2. La evaluación de un elemento de la red no está completa sino hasta que estén evaluadas todas las salidas del elemento.
3. La evaluación de la red termina cuando se han evaluado todas las salidas de todos los elementos de la red.

### Lista de instrucciones (*Instruction List, IL*)

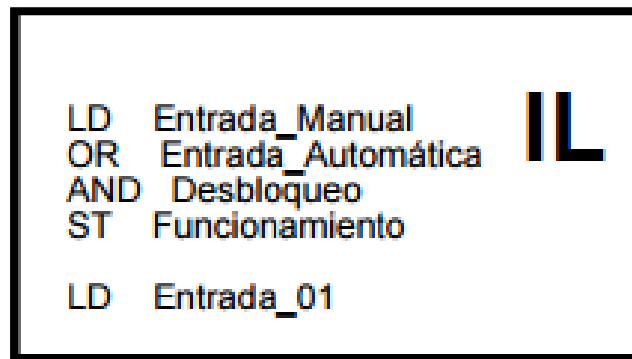


Figura 2.7 Ejemplo de una Lista de instrucciones

Lenguaje tipo texto, es un lenguaje ensamblador, donde cada línea de código obedece a una instrucción a ejecutar por el controlador programable. Cada sentencia o línea de código está compuesta por:

**Etiqueta: Operador/Función Operando**

- Etiquetas: son necesarias para que se puedan ejecutar los saltos del programa que se tienen en las líneas de código en el resto del fichero.
- Operador: indica la instrucción que se va a realizar.
- Operando: indica una constante o dirección con la que debe trabajar la operación.

### Modificadores y operadores en lenguaje IL

En el lenguaje IL se pueden utilizar los siguientes operadores y modificadores.

#### *Modificadores:*

- C en JMP, CAL, RET: la instrucción sólo se ejecuta si el resultado de la expresión anterior es TRUE.
- N en JMPC, CALC, RETC: la instrucción sólo se ejecuta si el resultado de la expresión anterior es FALSE.
- N en el resto: negación del operando (no del acumulador)

#### *Operadores*

A continuación aparece una tabla de todos los operadores en lenguaje IL con sus posibles modificadores y su correspondiente significado:

Operador	Modificadores	Significado
LD	N	Establecer resultado actual igual al operando
TE	N	Guardar resultado actual en la posición del operando
S		Ajustar el operando de Bool en TRUE cuando el resultado actual es TRUE
R		
AND	N, (	AND por bits
OR	N, (	OR por bits
XOR	(	OR exclusivo por bits
ADD	(	Suma
SUB	(	Resta
MUL	(	Multiplicación
DIV	(	División
GT	(	>
GE	(	>=
EQ	(	=
NE	(	<>
LE	(	<=
LT	(	<
JMP	CN	Saltar a la marca
CAL	CN	Llamar a Bloque de Funciones
RET	CN	Volver de la llamada a un Bloque de Funciones
)		Evaluar operación de repuesta

Tabla 2.2 Tabla de operadores IL. Tomado de [5]

## Lenguaje Texto estructurado (*Structured Text, ST*)

```

IF Data = "EOF" THEN
  FOR Index:=1 TO 128 DO
    X:=Read_Data(Datenfeld[index]);
    IF X > 2500 THEN Alarma:=TRUE;
  END_IF;
END_FOR;
END_IF;

```

ST

Figura 2.8 Ejemplo de un Texto estructurado

Lenguaje estructurado de alto nivel, originario de lenguajes como Pascal, Ada y C. Es una construcción sintáctica que al ser evaluada proporciona un valor, la evaluación se efectúa según una tabla de prioridades. Presenta los mismos elementos que el lenguaje IL, compuesto por operadores y operandos.

La programación en Texto Estructurado es apropiada para aplicaciones que involucran manipulación de datos, ordenamiento computacional y aplicaciones matemáticas que utilizan valores de punto flotante. Este lenguaje resulta muy para la definición de bloques de función complejos o procedimientos que no pueden ser expresados mediante lenguajes gráficos.

- **Tabla con los operadores ST por orden de fuerza de enlace.**

Operación	Símbolo	Fuerza de enlace
Entre paréntesis	(expresión)	Enlace más fuerte
Llamada a funciones	Nombre de función (lista de parámetros)	
Potenciar	EXPT	
Negar Formación de complemento	- NOT	
Multiplicar Dividir Módulo	* / MOD	
Sumar Restar	+ -	

<b>Comparar</b>	<, >, <=, >=	
<b>Igualdad</b>	=	
<b>Desigualdad</b>	<>	
<b>Bool AND</b>	AND	
<b>Bool XOR</b>	XOR	
<b>Bool OR</b>	OR	Enlace más débil

Tabla 2.3 Operadores de Lenguaje ST. Tomado de [6]

- **Tabla con las instrucciones utilizadas en ST**

<b>Tipo de instrucción</b>	<b>Ejemplo</b>
<b>Asignación</b>	A:=B; CV := CV + 1; C:=SIN(X);
<b>Llamada a un Bloque de Funciones y uso de la salida FB</b>	CMD_TMR(IN := %IX5, PT := 300);A:=CMD_TMR.Q;
<b>RETURN</b>	RETURN;
<b>IF</b>	IF D:=B*B;IF D<0.0 THEN C:=A;ELSIF D=0.0 THEN C:=B;ELSE C:=D;END_IF;
<b>CASE</b>	CASE INT1 OF1: BOOL1 := TRUE;2: BOOL2 := TRUE;ELSE BOOL1 := FALSE; BOOL2 := FALSE;END_CASE;
<b>FOR</b>	FOR J:=101;FOR I:=1 TO 100 BY 2 DO IF ARR[I] = 70 THEN J:=I; EXIT; END_IF;END_FOR;
<b>WHILE</b>	WHILE J<= 100 AND ARR[J] <> 70 DO J:=J+2;END_WHILE;
<b>REPEAT</b>	REPEAT J:=J+2;UNTIL J= 101 OR ARR[J] = 70END_REPEAT;
<b>EXIT</b>	EXIT;
<b>Instrucción vacía</b>	;

Tabla 2.4 Lista de instrucciones en el lenguaje ST. Tomado de [6]



## 2.3 Entornos de desarrollo para la programación de PLC

### 2.3.1 CoDeSys

#### ¿Qué es CoDeSys?

El término CoDeSys (*Controlled Development System*) es un acrónimo y significa sistema de desarrollo de controladores. Este programa fue creado y actualmente comercializado por la empresa de software alemán 3S-Smart de Soluciones de Software. Es un entorno de desarrollo para la programación de autómatas programables, conforme al estándar internacional IEC 61131-3.

Este software cuenta con los cinco lenguajes de programación (IL, ST, LD, FBD, SFC) definidos en el estándar IEC 61131-3, además de un editor de FBD libre. Es un editor orientado al lenguaje FBD donde las conexiones entre las entradas, salidas y los operadores se fijan automáticamente.

Más de 250 fabricantes de dispositivos de diferentes sectores industriales ofrecen sus dispositivos de automatización inteligente programable con la interfaz de programación CoDeSys, entre ellos ABB, Beckhoff, Wago, Moeller. Miles de usuarios finales en todo el mundo emplean CoDeSys para su trabajo diario en todo tipo de tareas de automatización. Una red mundial de asociados del sistema de CoDeSys ofrece tanto una amplia variedad de servicios para los usuarios CoDeSys como el apoyo a los usuarios finales, soporte, consultoría, formación, programación de aplicaciones o la integración de sistemas. [7]

#### Interfaz del programa

Mediante un ejemplo de programación se mostrará la interfaz del programa CoDeSys, de esta manera se pretende ampliar el conocimiento en este software que actualmente es muy conocido en los mercados de PLC's.

Ejemplo de programación: procedimiento de encendido y apagado de un motor. El objetivo es realizar el código en el lenguaje LD y visualizar su procedimiento en una interfaz Hombre-Máquina.

Procedimiento para realizar la programación:

Cuando se abre el software aparece la interfaz del programa (Figura 2.9), en el menú de opciones se debe seleccionar la opción Archivo/Nuevo.

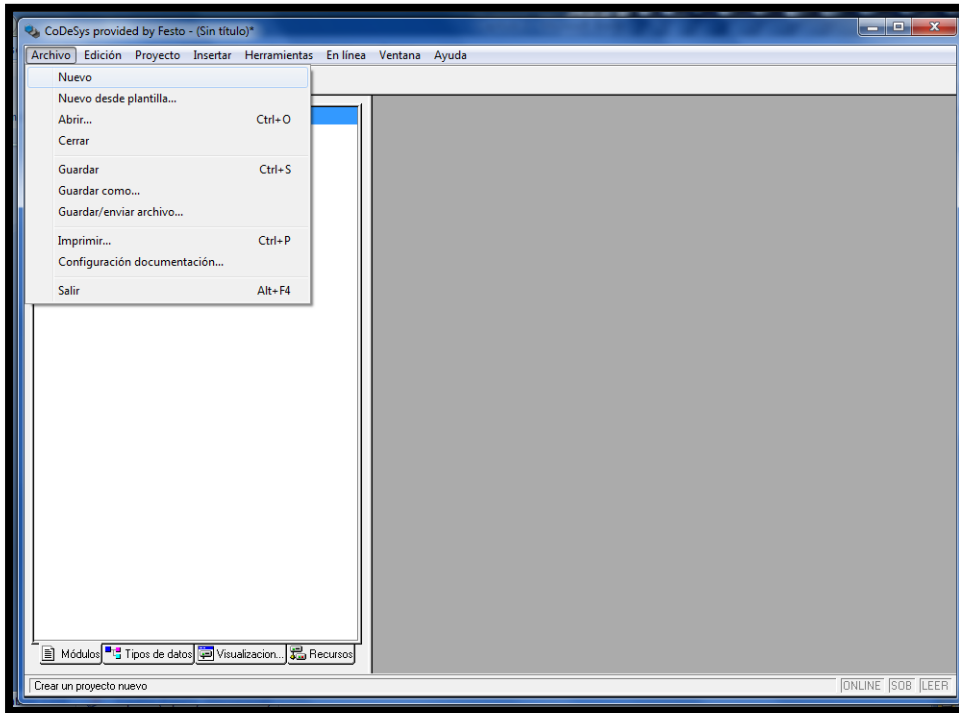


Figura 2.9 Interfaz de CoDeSys

Se observa un cuadro de dialogo para la configuración del PLC al cual se desea conectar, en este caso no se dispone de un PLC físico, además el objetivo de este capítulo es mostrar el manejo de la interfaz de trabajo de CoDeSys, por esta razón se escoge la opción *None*. (Figura 2.10)



Figura 2.10 Ventana de Configuraciones del sistema de destino

A continuación se abre una ventana de “Nuevo Módulo” (Figura 2.11), de ella se destacan varios aspectos:

Cuando se crea un módulo por primera vez, el software activa un POU predeterminado cuyo nombre siempre será PLC\_PRG, con la característica que es un POU de ejecución cíclica indefinida y sin condiciones, siempre lo ejecutara el sistema. Es similar a la marca especial SM0.1 en Step 7 Microwin.

Como se aprecia en la Figura 2.11, en esta ventana también escoge el tipo de modulo y el lenguaje de programación. Para el ejemplo el Tipo de modulo es Programa y el lenguaje grafico Diagrama de contacto (LD).

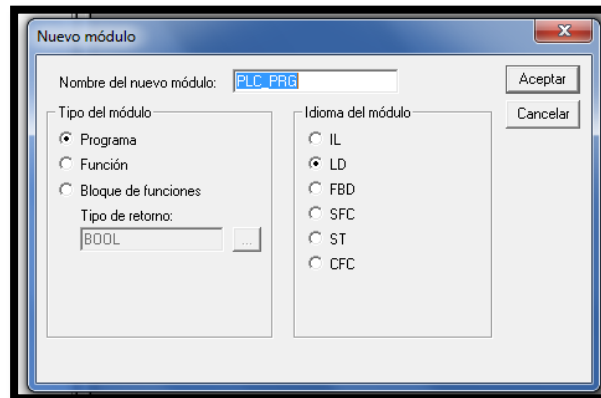


Figura 2 11 Ventana de Nuevo módulo

Posteriormente aparece el entorno de trabajo (Figura 2.12), este se puede dividir en cuatro partes:

1. Navegador de Objetos o POU's
2. Panel de herramientas.
3. Declaración de Variables.
4. Editor de programa.

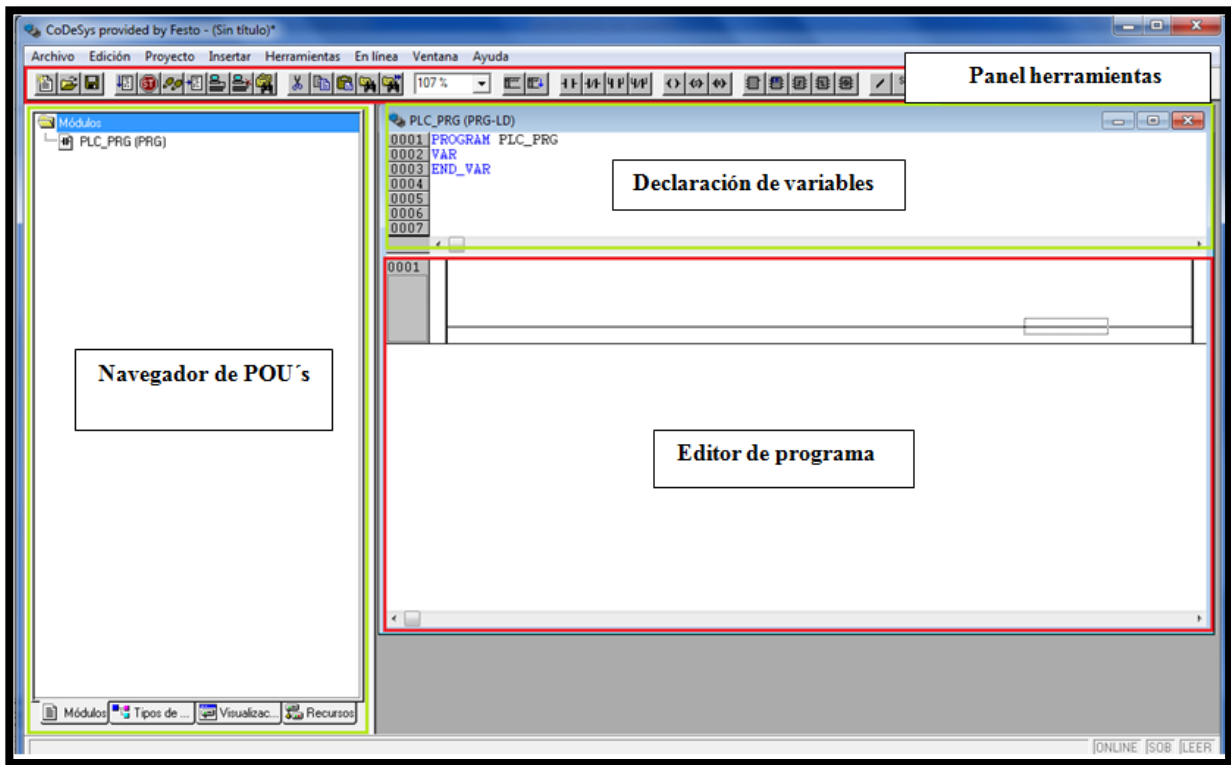


Figura 2.12 Entorno de trabajo

1. Navegador de Objetos o POU's:

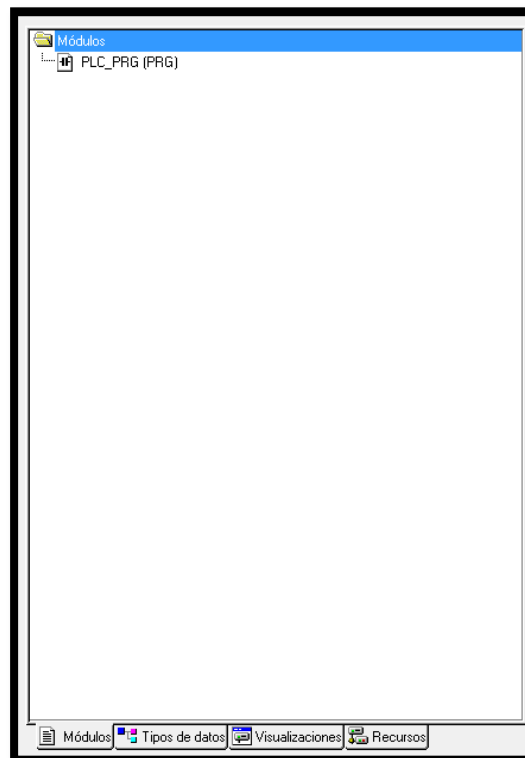


Figura 2.13 Navegador de POU's

Lo primero que se observa en el navegador, es una carpeta llamada Módulos, cada vez que se crea un programa, función o bloque de función, se ubicará en esta área. Para el ejemplo, aparece el programa predeterminado por el software, PLC\_PRG (Figura 2.13)

En la parte inferior del navegador se encuentra los menús siguientes:

- Tipos de datos.
- Visualizaciones: el software ofrece la posibilidad de visualizar las variables del proyecto. Además, una visualización puede utilizarse como superficie de mando exclusivo de un proyecto con el HMI (*Human Machine Interface*) de CoDeSys.
- Recursos: seleccionando este menú se aprecia la ventana que se muestra en la Figura 2.14

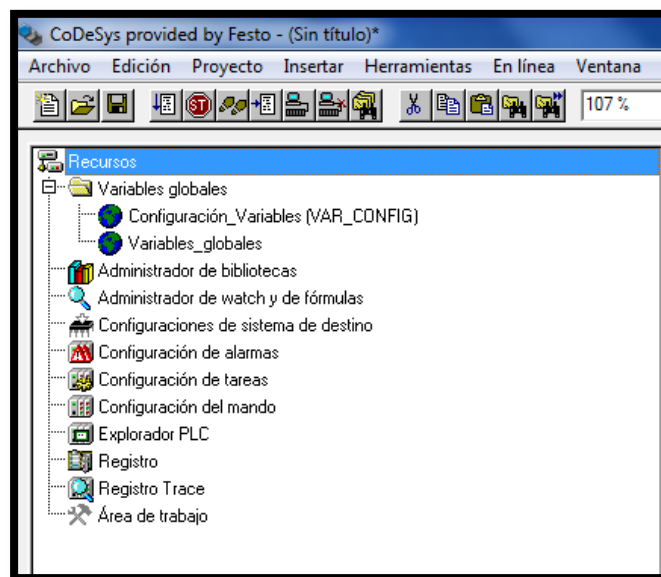


Figura 2.14 Opción Recursos

[8] Los recursos son necesarios para la configuración y organización del proyecto y para realizar un seguimiento de los valores de las variables:

- ✓ Variables globales que pueden utilizarse en todo el proyecto.
- ✓ Bibliotecas, que pueden enlazarse al proyecto a través del administrador de bibliotecas
- ✓ Registro lleva el control de las actividades online

- ✓ Configuración de alarma para la configuración del tratamiento de alarma en el proyecto
- ✓ Configuración del sistema de control para configurar su hardware
- ✓ Configuración de tareas para dirigir su programa mediante tareas
- ✓ Administrador de watch y de fórmulas para mostrar y preestablecer valores de variables
- ✓ Configuración del sistema de destino para seleccionar, y en caso necesario, terminar de configurar el sistema de destino
- ✓ Área de trabajo con una copia de las opciones de proyecto

## 2. Panel de herramientas:

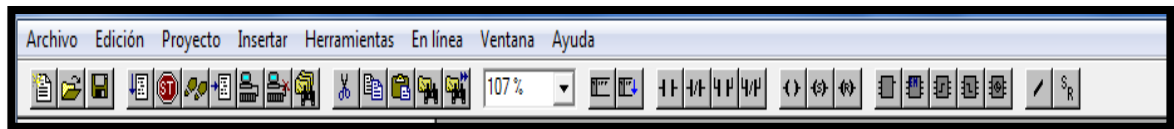


Figura 2.15 Panel de herramientas

En este panel (Figura 2.15) de izquierda a derecha se localizan las siguientes opciones: las opciones de documentos (nuevo documento, abrir y guardar documento), después las opciones referidas al autómata, entre ellas, cargar el código al PLC, parar la ejecución del código, estado de entradas y salidas, etc. Por último se encuentra un área reservada para los lenguajes gráficos, aquí se ubican los elementos de cada lenguaje.

En este ejemplo en el que se utiliza el lenguaje LD, se aprecia elementos típicos de este lenguaje como contactos, bobinas, funciones y bloques funcionales.

## 3. Declaración de Variables.



Figura 2.16 Declaración de variables locales

Zona para declarar Variables Locales, las variables declaradas aquí, se caracterizan por estar entre las instrucciones VAR y END\_VAR. (Figura 2.16). Este editor de texto aparece cuando se crea un módulo nuevo y en él solo aparecen las variables utilizadas en dicho modulo.

La sintaxis para la declaración de cualquier variable es:

```
[Nombre] [ : ] [Tipo] [ ; ]
```

Figura 2.17 Sintaxis para declarar una variable

En el ejemplo se omitirá el uso de las variables locales. Ya que, como se mencionó en el apartado de “Navegador de POU’s”, para realizar una HMI, es indispensable el uso de variables globales, es decir que se puedan utilizar en todo el proyecto.

Para la declaración de las Variables Globales, una vez se ha ubicado el menú Recursos, se selecciona la opción Variables globales/Variables\_globales.

En la Figura 2.18 se observa que para el ejemplo, se declararon tres variables globales: START, STOP y MOTOR, todas de tipo booleano.

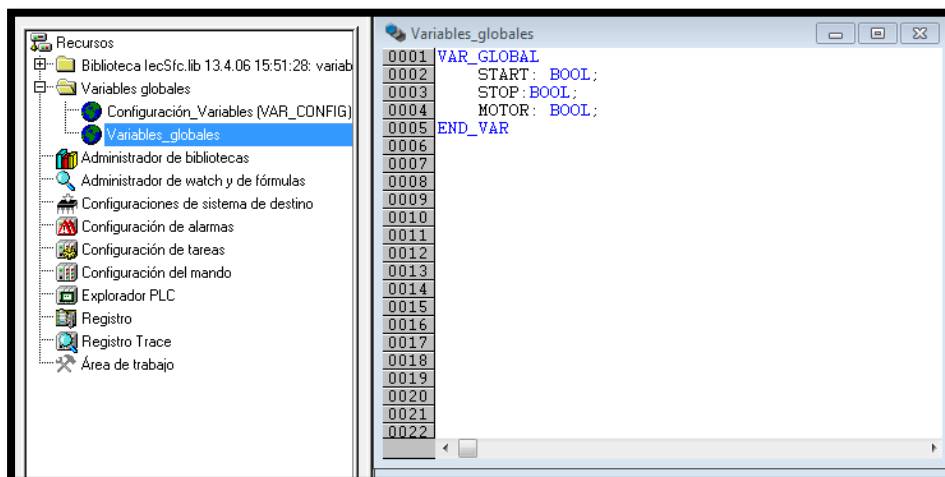


Figura 2.18 Declaración de las variables globales

#### 4. Editor de programa.

Para realizar el código del ejemplo, se seleccionó la zona de trabajo indicada (con doble click), seguidamente se escogió del panel de herramientas los contactos y bobinas necesarios (Figura 2.19).

El software CoDeSys presenta muchas similitudes con Step 7 MicroWin, lo que facilita la programación en él.

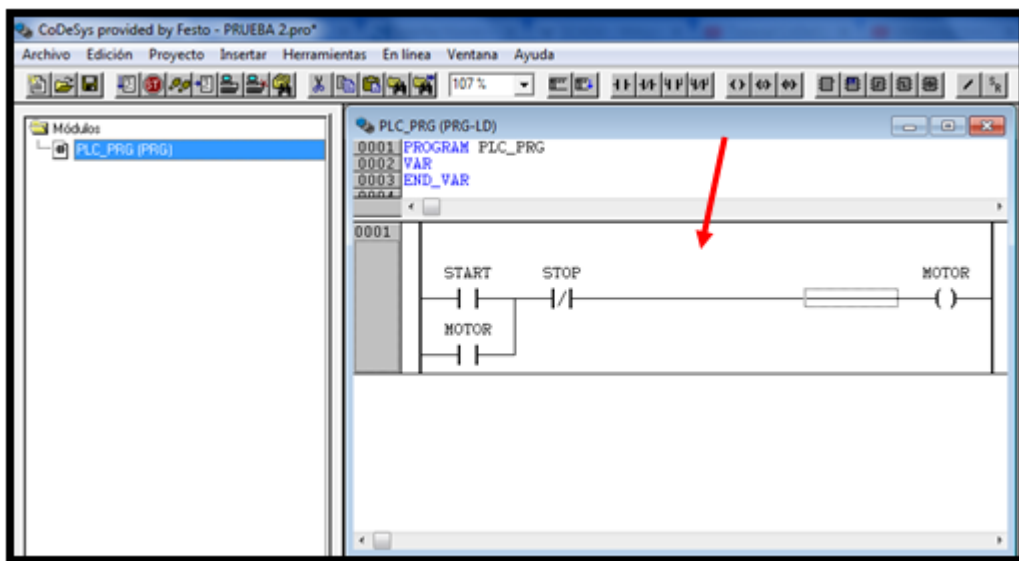


Figura 2.19 Editor del programa



Para crear la interfaz Hombre-Máquina, se debe ir al menú de Visualizaciones, en esta opción se selecciona Insertar objeto (Figura 2.20).

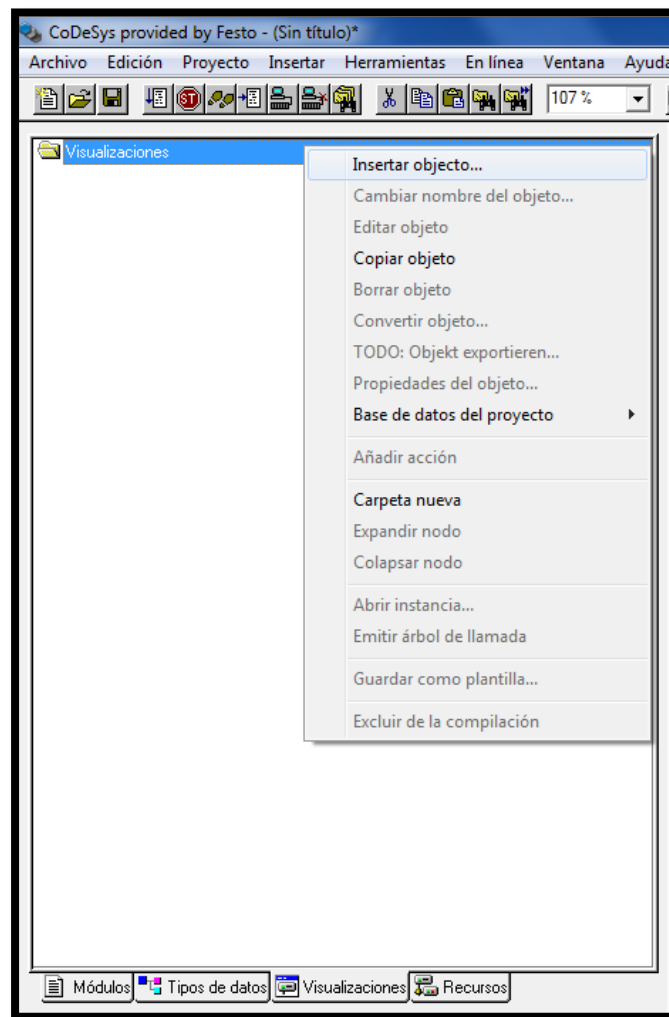


Figura 2.20 Menú de Visualizaciones

Como se observa en la Figura 2.21, aparece una ventana donde se debe indicar la aplicación de control a utilizar, para ello CoDeSys cuenta con el sistema SCADA (*Supervisory Control And Data Acquisition*)

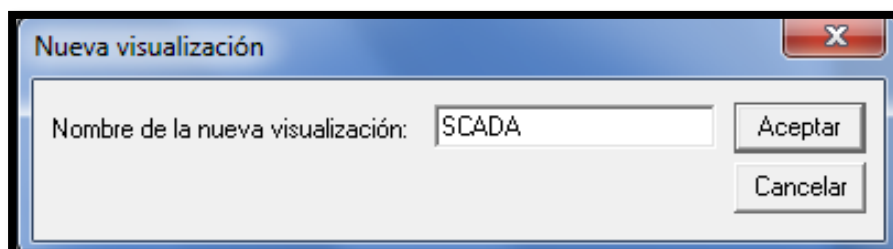


Figura 2.21 Ventana Nueva de visualización

A continuación se observa la interfaz de SCADA (Figura 2.22).

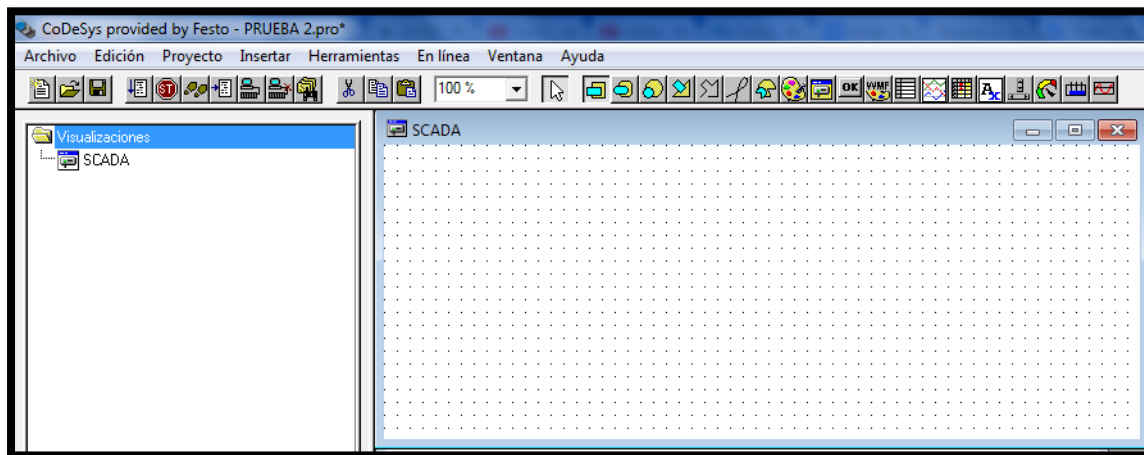


Figura 2.22 Interfaz SCADA

Para realizar los botones de START y STOP, se debe seleccionar del panel de herramientas el icono con las letras "OK" y para representar el motor el icono del círculo, tal como se muestra en la Figura 2.23.

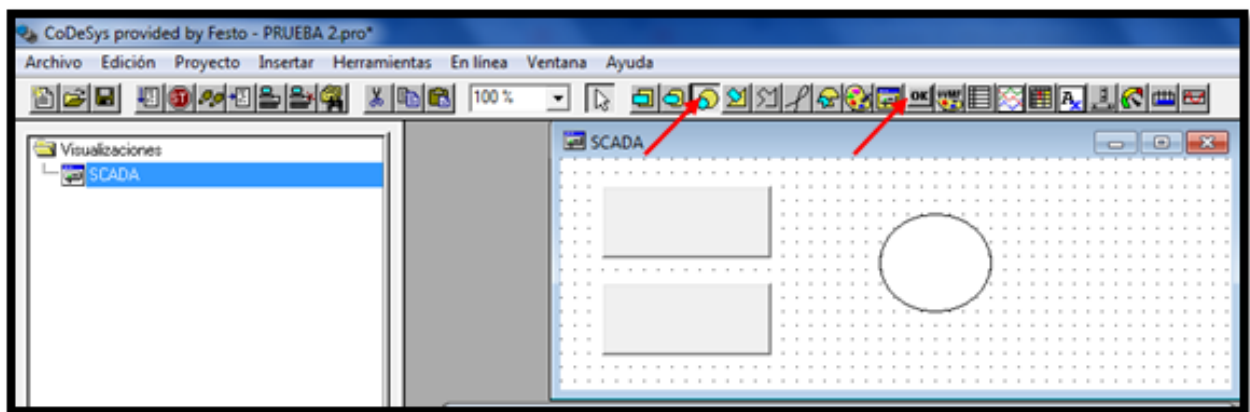


Figura 2.23 Entorno de trabajo SCADA

Seleccionando el objeto creado (con doble click), aparece un cuadro dialogo donde se configura el elemento. (Figura 2.24).

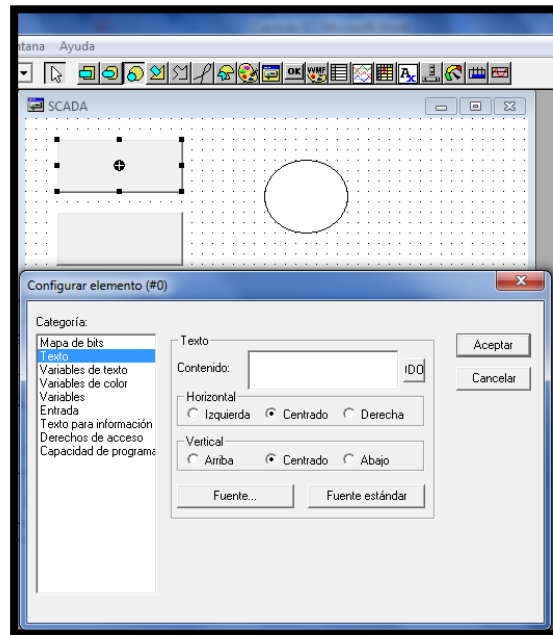


Figura 2.24 Configurar Elemento

Configuración del nombre del elemento (Figura 2.25).

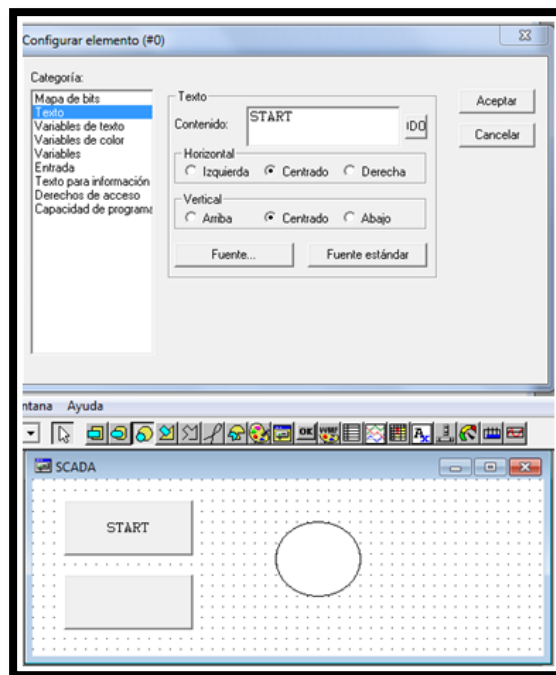


Figura 2.25 Configurar nombre del elemento

Configuración del color del elemento (Figura 2.26).

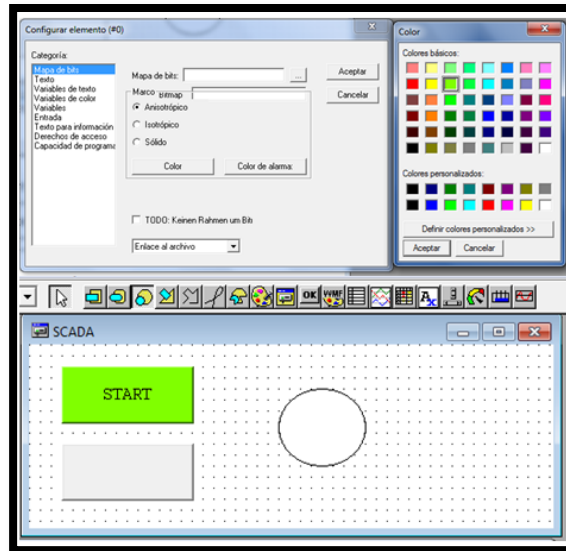


Figura 2.26 Configurar color del elemento

Asignación de la variable al elemento (Figura 2.27).

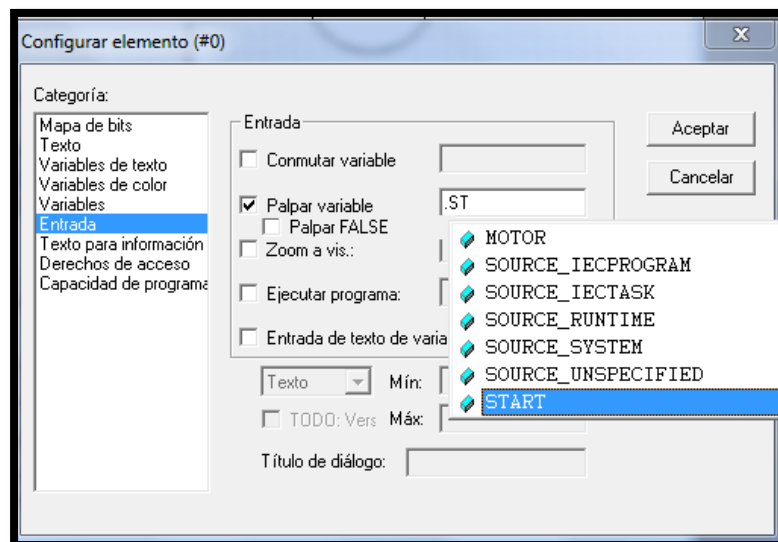


Figura 2.27 Asignación de la variable al elemento

Este proceso se repite para los tres elementos, el resultado final se aprecia en la Figura 2.28.

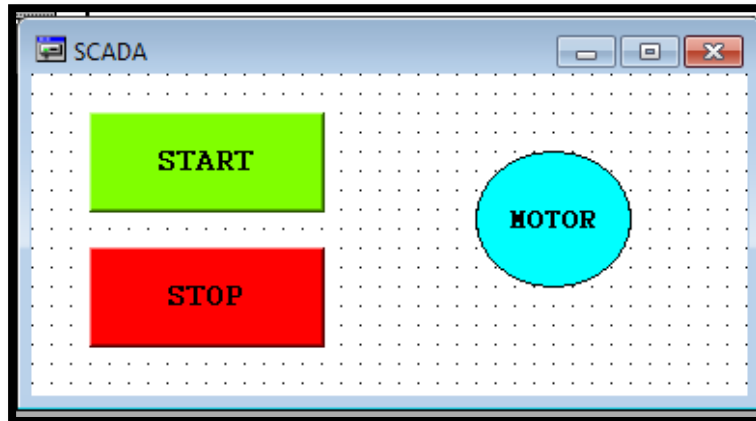


Figura 2.28 Resultado del ejemplo

Para activar la simulación del proceso, del menú principal, se selecciona la opción En Línea/Simulación (Figura 2.29).

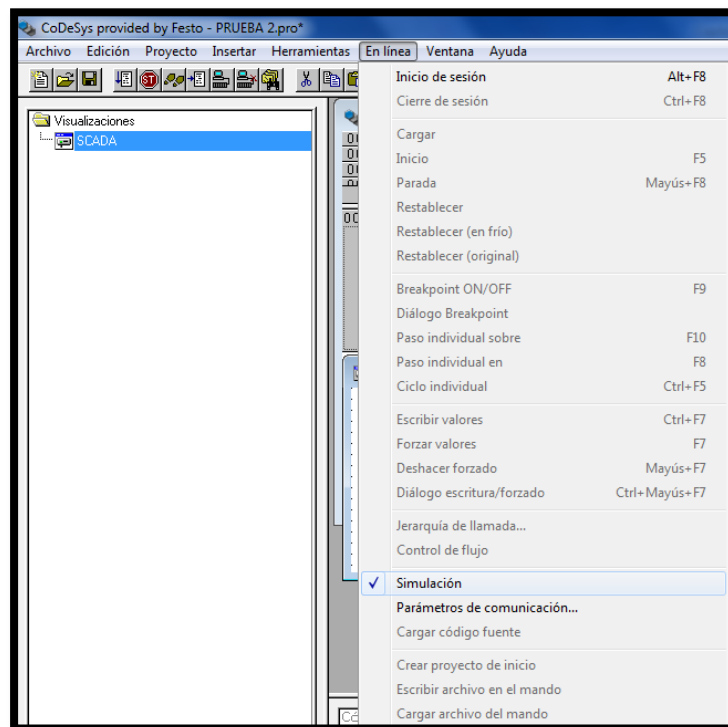


Figura 2.29 Opción En Línea

El siguiente paso, es activar del panel de herramientas las opciones Inicio de sesión y Estado de los elementos (Figura 2.30).

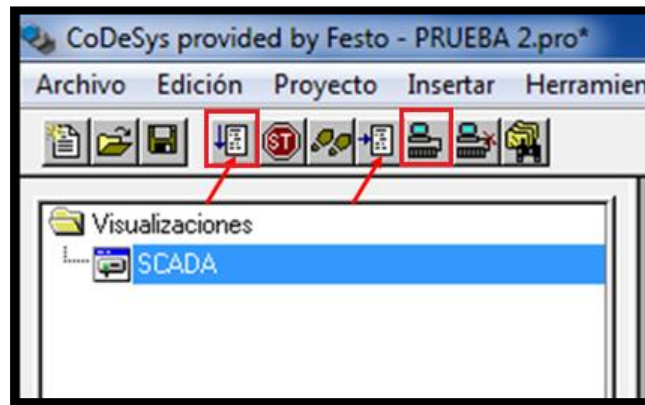


Figura 2.30 Opciones Inicio de sesión y Estado de los elementos

Con este procedimiento se logra el funcionamiento del código propuesto. Al pulsar los botones de STAR y STOP se comprueba el funcionamiento del motor (Figura 2.31).

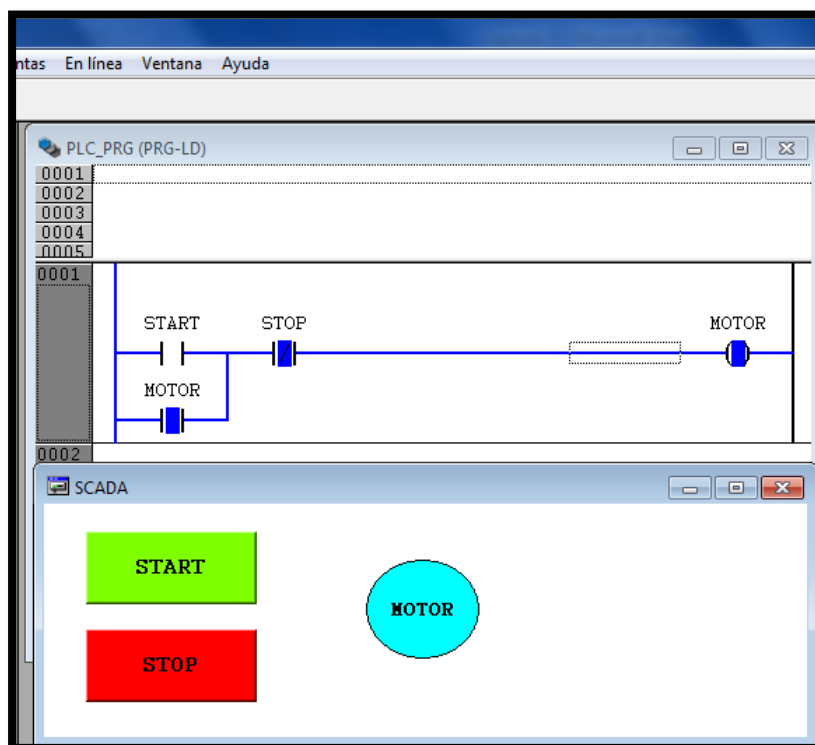


Figura 2.31 Comprobación del funcionamiento

### 2.3.2 STEP7- Microwin

#### ¿Qué es STEP7- Microwin?

Es un entorno de programación para autómatas programables creado y comercializado por la empresa Siemens, fue una de los primeros programas hechos por esta empresa adaptados a las especificaciones de la normativa estándar internacional IEC 61131-3. Microwin es un software especialmente diseñado y dotado de todas las herramientas para la programación de autómatas de la gama S7 – 200. Cuenta con tres lenguajes de programación:

- Diagrama de funciones **FUP** (*Funktions Plan*).
- Diagrama de contactos **KOP** (*Kontakt Plan*)
- Lista de instrucción **AWL** (*Anweisungs Liste*)

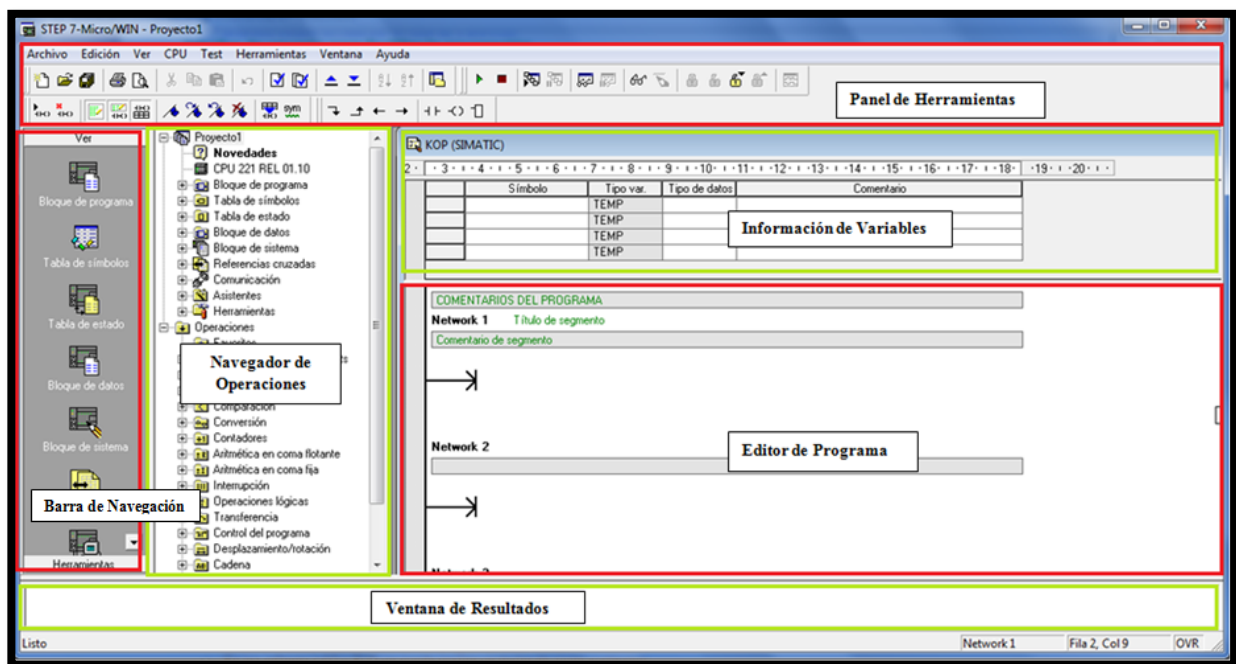


Figura 2.32 Interfaz de Step 7 Microwin

#### Uso de Step 7 Microwin en este proyecto

La Universidad de la Laguna, para la formación de los alumnos en materia de Automatización posee un convenio con la empresa Siemens. En el laboratorio de Automatización “Profesor Lorenzo Moreno Ruíz”, ubicado en la primera planta de la facultad de Ingeniería Informática, dispone de autómatas programables de la gama S7-200 y

S7-300 de Siemens al igual que en los ordenadores se encuentran instalados los entornos de programación específicos para cada autómeta.

Por este motivo para cumplir uno de los objetivos de este proyecto, la automatización de la Estación de acarreo de piezas, se utilizó el software Step 7 Microwin. Durante todo el curso las alumnas autoras de este proyecto, contaron con el acceso libre al laboratorio para realizar el código de automatización, verificar su funcionamiento tanto en el autómeta S7-200 como en estación real.

El trabajo realizado en el laboratorio se puede observar en los Anexos I y II de este documento, corresponden a los códigos de Automatización de la Estación de acarreo de piezas en los lenguajes de programación diagrama de contactos y lista de instrucciones.



## Capítulo 3. Descripción de la planta FESTO

### 3.1 Situación de la planta industrial FESTO

La planta con la que se trabaja en el presente proyecto se encuentra en el Laboratorio Profesor Lorenzo Moreno Ruíz, localizada en la primera planta de la facultad de Ingeniería Informática. Ésta se trata de un conjunto de estaciones que simulan un proceso industrial.

- La planta industrial Festo se divide consta de cinco estaciones:
- Estación 0: Se encarga de almacenar y distribuir las piezas de la planta.
- Estación 1: Test, mide la longitud de las piezas.
- Estación 2: Procesado de piezas.
- Estación 3: Acarreo de piezas.
- Estación 4: Clasificación de las piezas.

Cada estación se controla con Autómatas Programables o PLC (*Programmable Logic Controller*) de Siemens que pueden ser el S7-200 o el S7-300. La estación tratada para el proyecto será la tercera, controlada por un autómata S7-200.

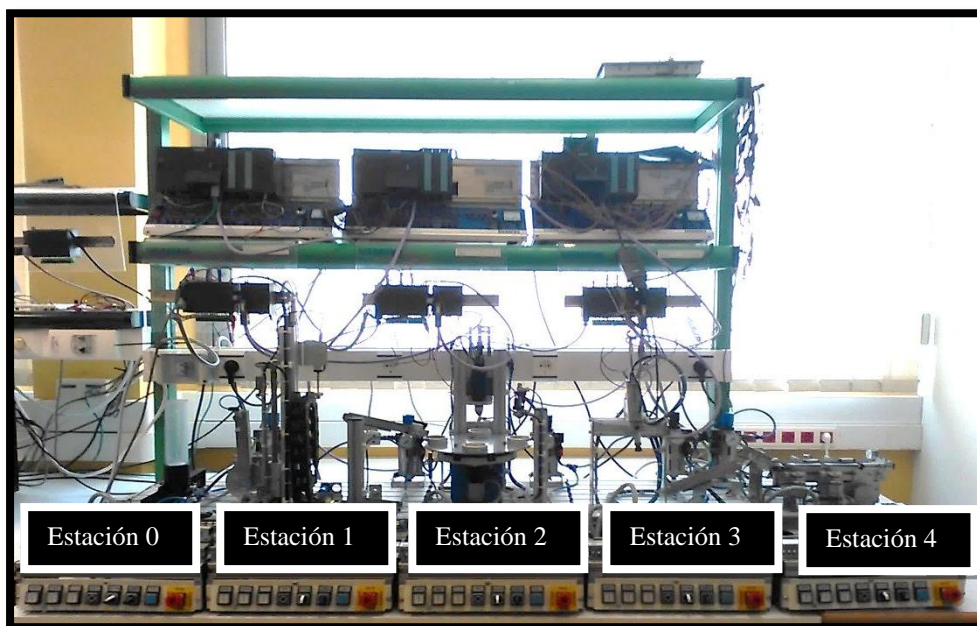


Figura 3.1 Estaciones de la planta Festo

### 3.2 Estación de acarreo de piezas

El objetivo de la estación de acarreo de piezas es transferir las piezas que llegan de la estación anterior (estación 2) y clasificarlas como “no defectuosas” o “defectuosas”. En caso de que la pieza se considere buena (no defectuosa) se enviará a la siguiente estación (clasificación de las piezas). En caso contrario la pieza se enviará a un almacén de piezas defectuosas.

### 3.3 Descripción de los elementos de la estación

En la estación de acarreo de piezas se pueden distinguir dos subsistemas, que son:

- El brazo de transferencia
- El almacén de piezas defectuosas

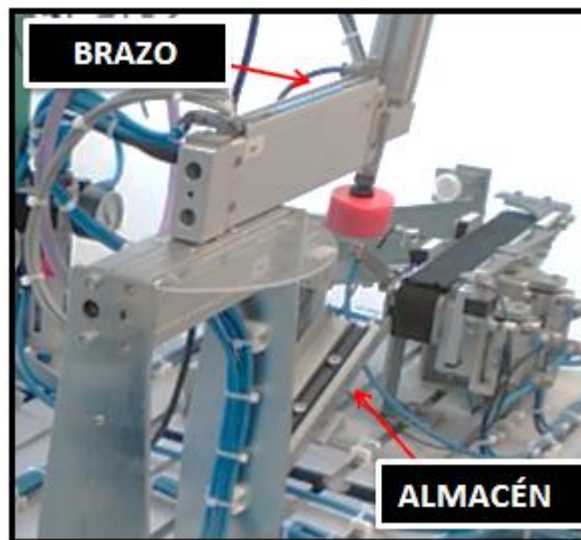


Figura 3.2 Subsistemas de la estación

#### 3.3.1 Brazo de transferencia

El brazo de transferencia está compuesto por un sistema de rotación, un elemento horizontal extensible, y un elemento portador.

- Sistema de rotación. Es el que permite rotar el elemento extensible horizontal de derecha a izquierda ( $180^\circ$ ) para situarlo sobre la estación previa o sobre la estación posterior.

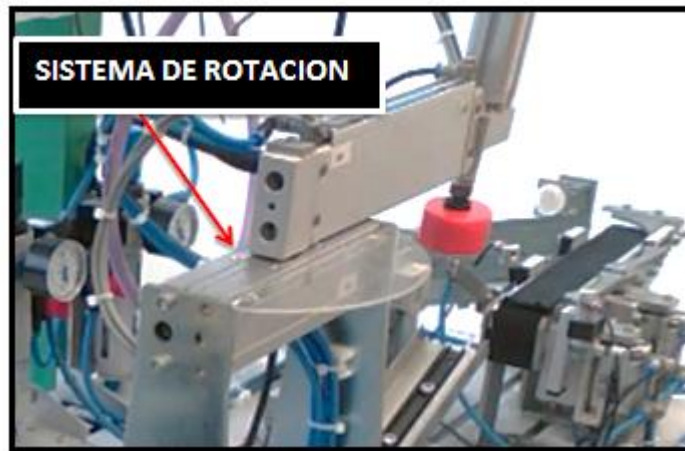


Figura 3.3 Sistema de rotación

- Elemento extensible horizontal. Éste puede estar extendido o retraído en la dirección horizontal. En uno de sus extremos se sitúa el eje de rotación solidario con el sistema de rotación
- Elemento portador. El elemento portador está en un extremo del elemento horizontal y en posición perpendicular. El elemento portador (elemento vertical) es extensible, esto le permite alcanzar la pieza que se quiere transportar y recogerla de la estación previa o depositarla en la estación posterior o al almacén de piezas defectuosa (dependiendo del caso). El elemento portador dispone de un sistema succionador en su extremo inferior que permite fijar la pieza suspendida de él y así transportarla hasta su destino final.

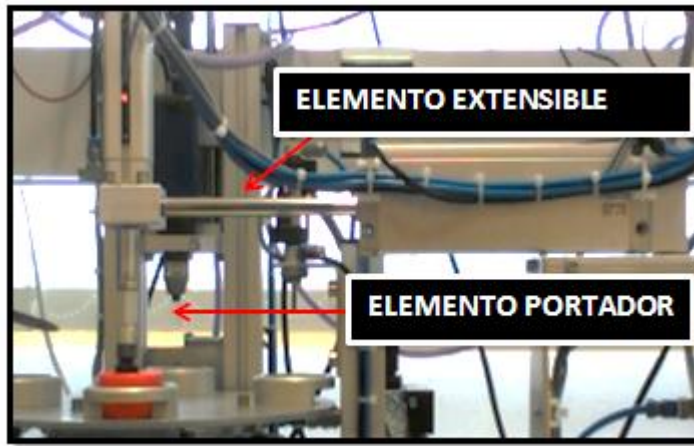


Figura 3.4 Elemento horizontal y portador

### 3.3.2 Almacén de piezas defectuosas

Se sitúa al lado del brazo de transferencia, en el extremo al que sigue la estación de destino. Se trata de un ‘tobogán’ cuyo extremo más elevado es alcanzable por el brazo de transferencia cuando el elemento extensible horizontal está retraído y apuntando hacia la estación posterior.

## 3.4 Sensores y Actuadores de la planta

### 3.4.1 Sensores

Brazo giratorio: dispone de sensores finales de carrera para determinar si él se encuentra en la estación previa o en la estación posterior.



Figura 3.5 Sensores del brazo giratorio

Elemento horizontal: dispone de sensores finales de carrera para determinar si el cilindro está extendido o retraído.



Figura 3.6 Sensores del elemento horizontal

Para el elemento vertical se emplean sensores finales de carrera, uno se activa cuando el elemento vertical está extendido, y el otro cuando está retraído.

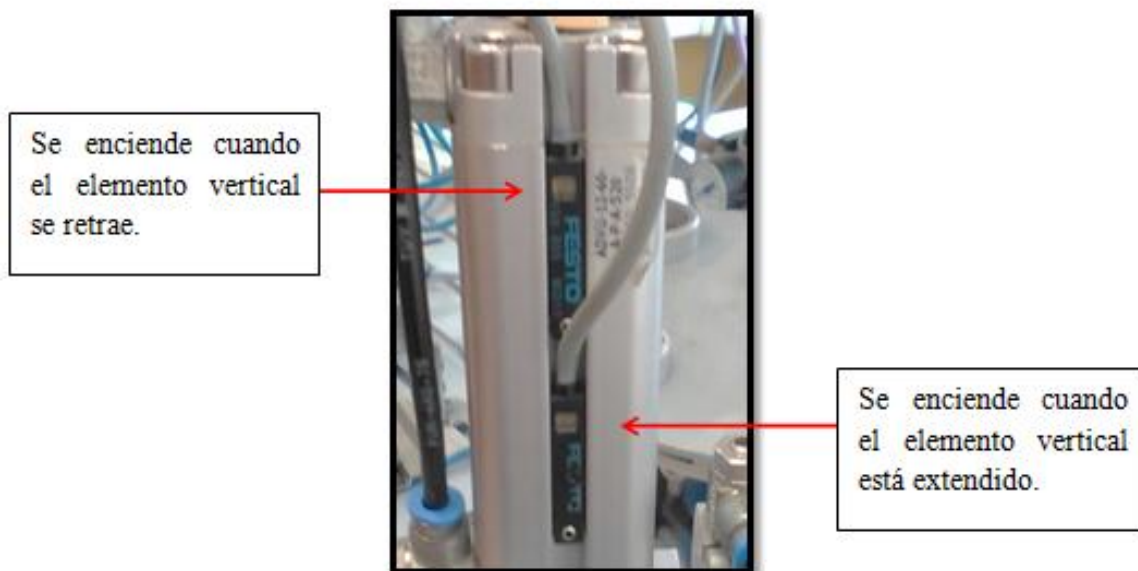


Figura 3.7 Sensores del elemento vertical

### 3.4.2 Actuadores

- Actuador rotatorio: se trata de un cilindro de doble efecto que acciona un piñón para transformar el movimiento lineal de un pistón en un movimiento circular que se aplica sobre un eje. Este sistema tiene dos entradas de aire, una en el extremo izquierdo y otra en el extremo derecho, controladas ambas por un sistema de válvulas.
- Elemento extensible horizontal: también se basa en un cilindro de doble efecto que, mediante una válvula, nos permite la extensión o retracción del mismo.

- Elemento extensible vertical: se implementa con un cilindro de doble efecto controlado por una válvula de un solo solenoide. Al activar el accionador solenoidal de la válvula, el cilindro se mueve en dirección vertical hacia abajo. Al desactivar el accionador de la válvula, ésta pasa a su segunda posición inyectando aire comprimido en el extremo opuesto del cilindro y haciendo que éste se mueva hacia arriba.



Figura 3.8 Válvula que activa al elemento vertical

- Ventosa: el sistema de vacío también se controla mediante una válvula. Su cometido es inyectar o dejar de inyectar aire comprimido en el sistema de vacío. Cuando se inyecta aire se consigue el efecto succionador.

### 3.5 Funcionamiento de la estación

A continuación se hará una descripción de la secuencia que debe realizar la estación de acarreo:

1. Las piezas llegan a la tercera estación de una en una por medio de una mesa giratoria proveniente de la estación 2. Inicialmente el brazo que recoge la pieza debe estar en la estación anterior con los pistones horizontal y vertical retraídos y la ventosa que succiona la pieza para poder levantarla debe estar en off.



Figura 3.9 Brazo en posición inicial

2. Cuando se detecte la presencia de una pieza en la estación 2, se acciona el pistón para extender el elemento horizontal del brazo.

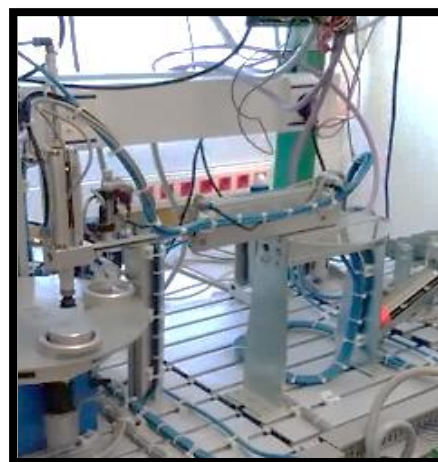


Figura 3.10 Elemento horizontal extendido

3. A continuación se recoge la pieza. Para ello se extiende el elemento vertical y la ventosa se activa (ON).

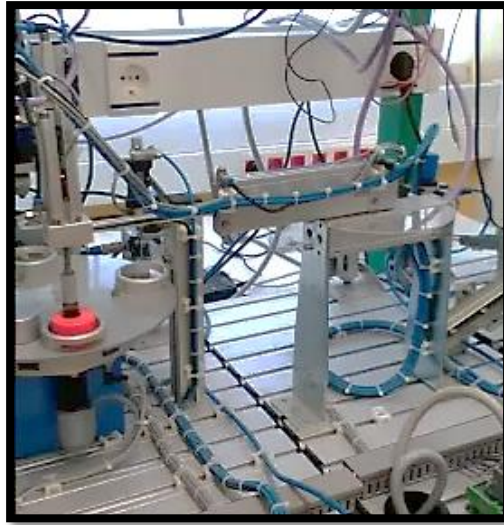


Figura 3.11 Pieza en ventosa

4. Cuando la pieza esté en la ventosa, se debe retraer el elemento vertical, luego el horizontal para llevar el brazo a la posición posterior.

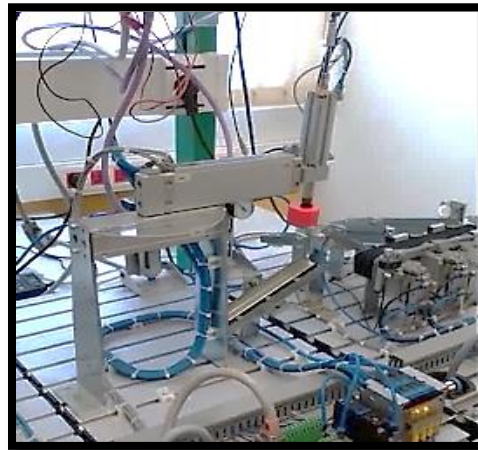


Figura 3.12 Brazo en posición posterior

5. Cuando se recoge la pieza, ésta puede ser clasificada como “no defectuosa” o “defectuosa”.



- En caso de que la pieza se clasifique como “no defectuosa”, se extiende el elemento horizontal, luego el vertical y se desactiva la ventosa. De esta manera la pieza se queda en la estación de clasificación de piezas (estación 4).



Figura 3.13 Pieza no defectuosa en estación 4

- Una vez que se deja la pieza en la estación 4, se retrae el elemento vertical y luego el horizontal.

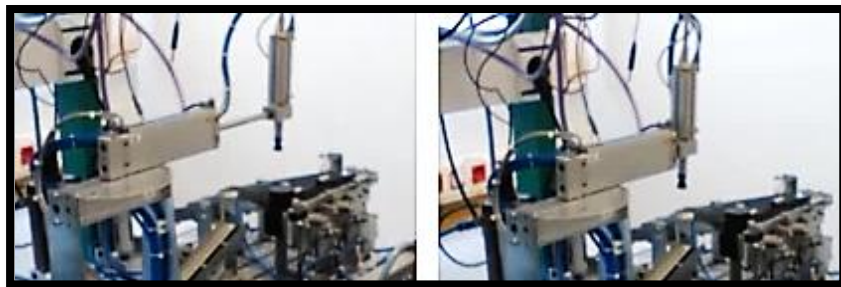


Figura 3.14 Retracción de elementos vertical y horizontal

- En caso de que la pieza se clasifique como “defectuosa”, se extiende elemento vertical y se desactiva la ventosa para dejar la pieza en el almacén.

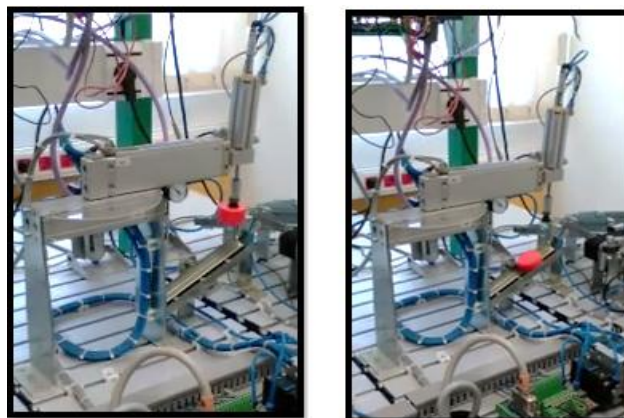


Figura 3.15 Pieza defectuosa en almacén

6. Una vez que se haya terminado uno de los dos procesos anteriores, el brazo debe volver a la estación previa y a las condiciones iniciales hasta que haya una nueva pieza.
7. En caso de que ocurra un incidente y la pieza se caiga antes de llegar a su destino, el brazo debe volver a la posición inicial.
8. Si produce una parada de emergencia, el sistema regresará de inmediato al estado inicial.

Además el sistema quedará detenido hasta que el operario vuelva a poner en marcha la estación. Para informar de que se pulsó la parada de emergencia, se encenderá una luz intermitente.

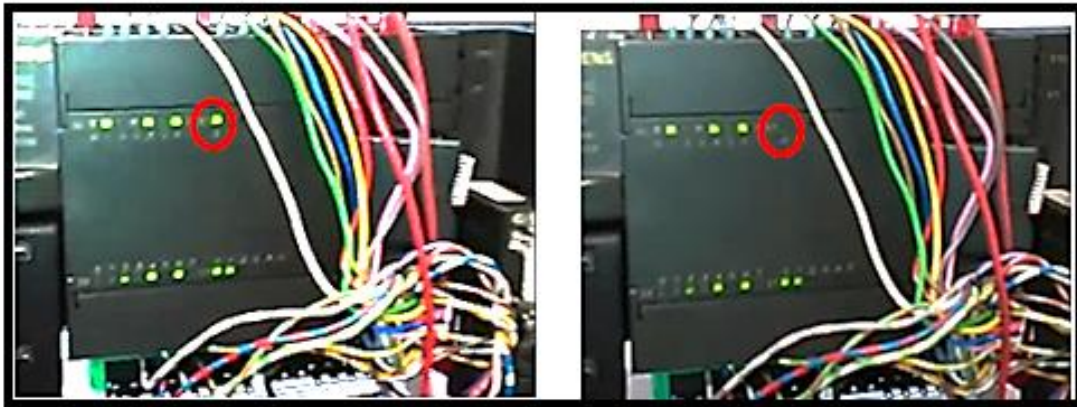


Figura 3.16 Luz de parada de emergencia

### 3.6 Conexión con el autómata

Como se ha mencionado anteriormente, el autómata que se utiliza para la estación de acarreo de piezas es el S7-200. A continuación se muestra el conexionado entre los sensores y las entradas del autómata, así como la de los actuadores y las salidas del mismo.

Entrada	Sensor
I0.1	Brazo girado en estación posterior.
I0.2	Brazo girado en estación previa.
I0.3	Elemento horizontal extendido.
I0.4	Elemento horizontal retraído.
I0.5	Elemento vertical extendido.
I0.6	Elemento vertical retraído.
I1.0	Pulsador de emergencia.
I1.1	Start.
I1.5	Pieza defectuosa.

Tabla 3.1 Entradas y Sensores del autómata

Salida	Actuadores
Q0.1	Retraer el elemento horizontal.
Q0.2	Extender el elemento horizontal.
Q0.3	Girar hacia estación posterior
Q0.4	Girar hacia estación previa.
Q0.5	Vacío ON.
Q0.6	Vacío OFF
Q0.7	Bajar/subir elemento vertical.

Tabla 3.2 Salidas y actuadores del autómata

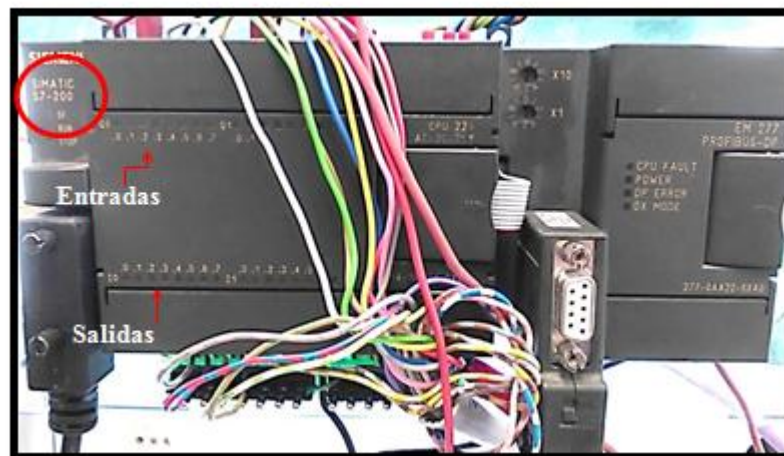


Figura 3.17 Autómata S7-200

## Capítulo 4. Simulación de estación de acarreo de piezas de la planta Festo

### 4.1 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 en el sistema real.

El motivo por el cual se realiza el modelado 3D y la simulación de la estación de acarreo de piezas, es para aportar un recurso de aprendizaje a los alumnos de la asignatura de automatización del grado de ingeniería electrónica industrial y automática de la universidad de la Laguna. Actualmente el laboratorio de automatización cuenta con una sola planta Festo para muchos alumnos y en horario limitado, lo que dificulta el aprendizaje.

### 4.2 Estudio del Software especializado para la simulación

El objetivo es encontrar un programa que realice de manera sencilla el modelado 3D y la simulación de la Estación de acarreo de piezas. 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:

#### 4.2.1 VariCAd

Software CAD 3D/2D creado especialmente para la ingeniería mecánica. En la interfaz de usuario se construye modelos 3D que fácilmente se convierten en documentación 2D, por ejemplo, los planos de cada pieza. Cabe destacar que cualquier cambio en los modelos 3D se actualiza en todos sus formatos.

Este programa cuenta con una biblioteca de formas básicas 3D, donde el elemento elegido se modifica editando sus dimensiones. Los sólidos pueden ser creados también por el perfil rotación, extrusión o herramientas más complejas.

Las piezas se definen con la fijación de parámetros y uso de restricciones geométricas, además cuenta con herramientas para depósitos, tuberías, láminas de metal, pruebas de choque, soporte de ensamblaje, cálculos de la parte mecánica y herramientas para trabajar con listas de materiales.

[9] El sistema contiene:

- modelado 3D
- Dibujo 2D y edición
- Soporte opcional de Parámetros
- Soporte opcional de Restricciones Geométricas
- Pruebas de choque
- Asambleas 3D
- Desarrollo de superficie
- Librería estándar de piezas mecánicas
- Los cálculos de secciones de los objetos en 3D o 2D
- Soporta STEP, STL, IGES, DWG y DXF.

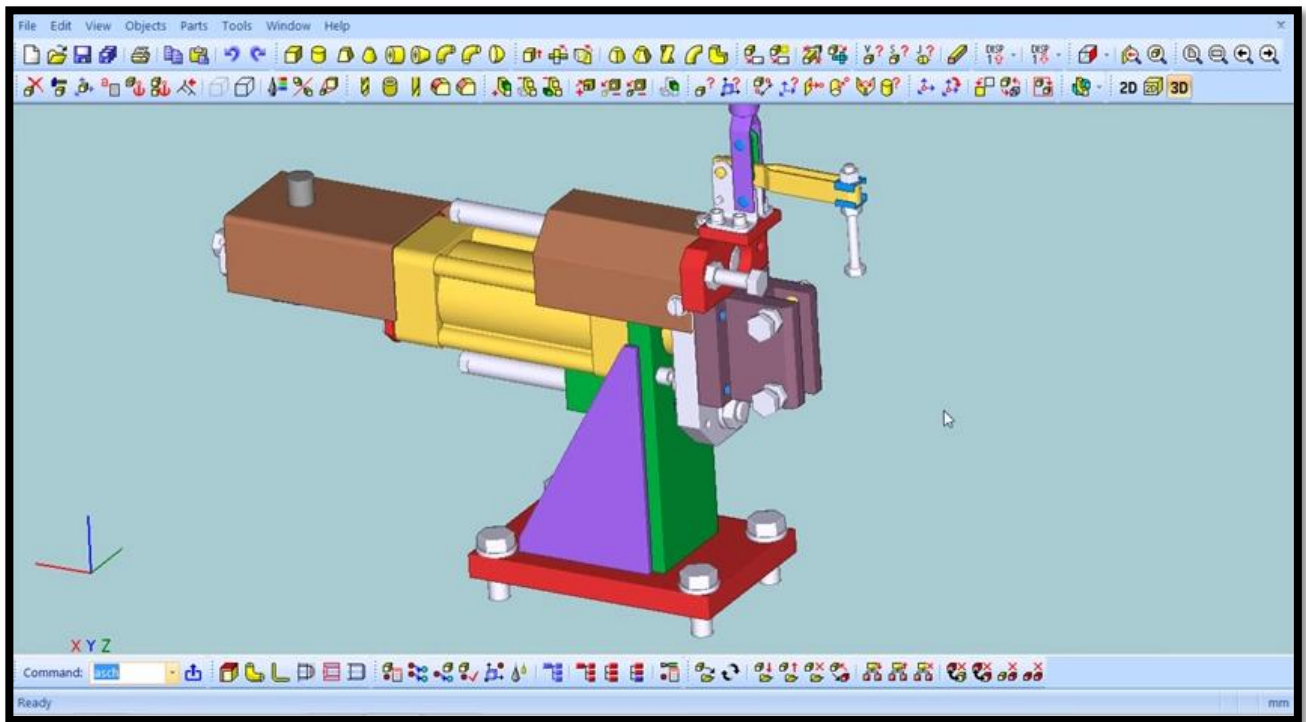


Figura 4. 1 Interfaz VariCAD

#### 4.2.2 SolidWorks

Software CAD 3D/2D para el desarrollo de proyectos industriales, el entorno de desarrollo permite crear un proyecto desde el nivel de boceto hasta cálculo dinámico de fluidos, comprobación de rotura, de rotación de piezas, de ensamblaje, obtención de planos, etc.

Posee modelado paramétrico con lo que facilita añadir o modificar las partes o dimensiones del modelo cuando el resultado no es el deseado, modificándose a tiempo real todos los planos y ensamblajes que se hayan generado de la pieza en cuestión.

Para la simulación del movimiento del modelo 3D, cuenta con el entorno “estudio de movimiento”. En él, las animaciones se realizan modificando las relaciones mecánicas entre sus piezas así como sus relaciones virtuales entre planos, todo en una escala de tiempo que el usuario puede modificar a su criterio para obtener el movimiento esperado del modelo.

Entre los paquetes de software que ofrece solidworks se encuentran SolidWorks Standard, SolidWorks Professional, SolidWorks Premium. Este último es una versión mejorada de solidWorks Profesional, que destaca por tener una nueva aplicación “Solidworks Motion” que añade al entorno de animación el cálculo físico real de las fuerzas y movimientos de un ensamblaje tal cual se movería en la realidad.

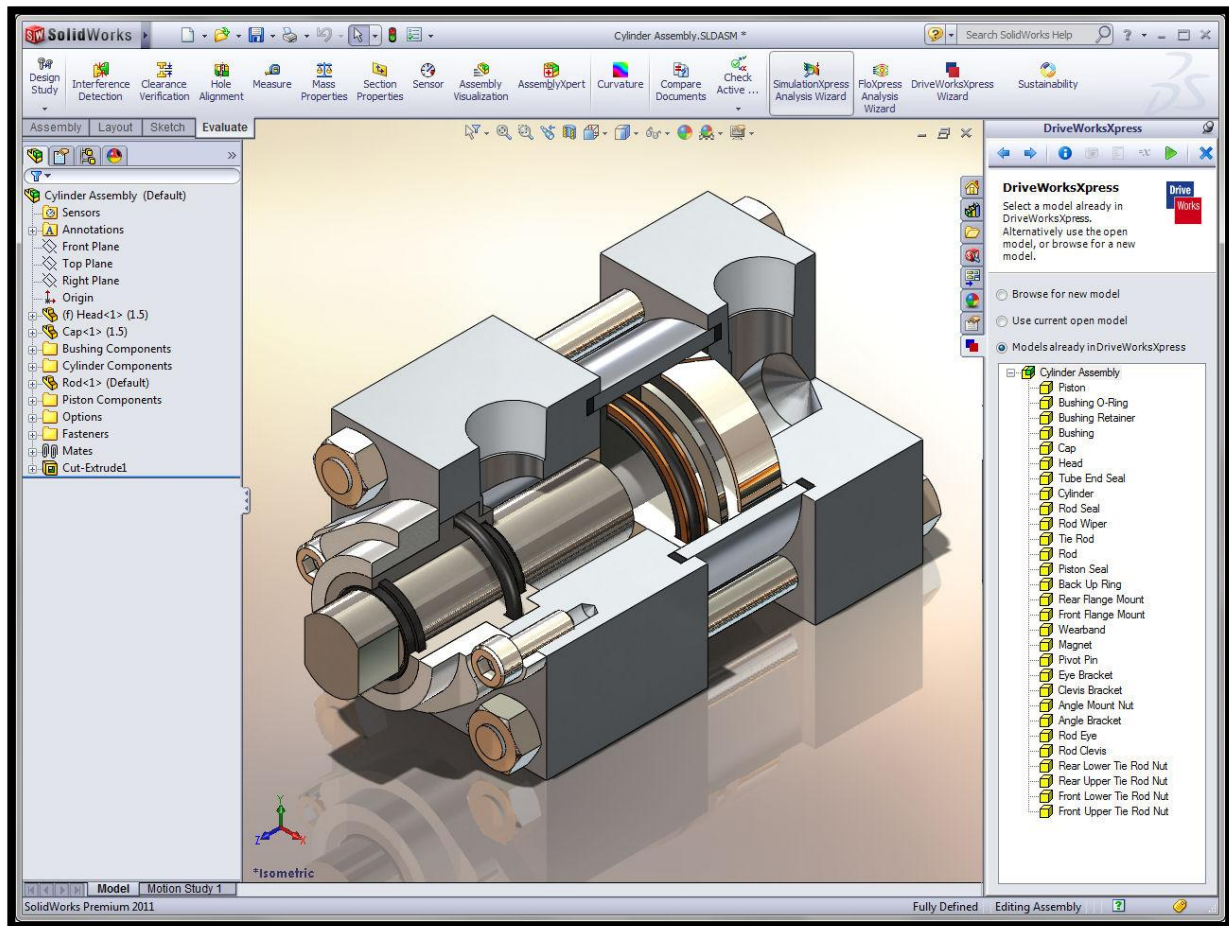


Figura 4.2 Interfaz SolidWorks

### 4.2.3 CATIA

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

Este programa está orientado al diseño avanzado de proyectos, su aplicación principal es el modelado avanzado de sólidos, superficies, ensamble, 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 final de animación. Incluye ensamblaje y planos en 2D.

El movimiento del modelo 3D, se realiza en el entorno “Kinematics Simulation”. La animación se logra modificando las restricciones de cada pieza en una escala de tiempo determinada.

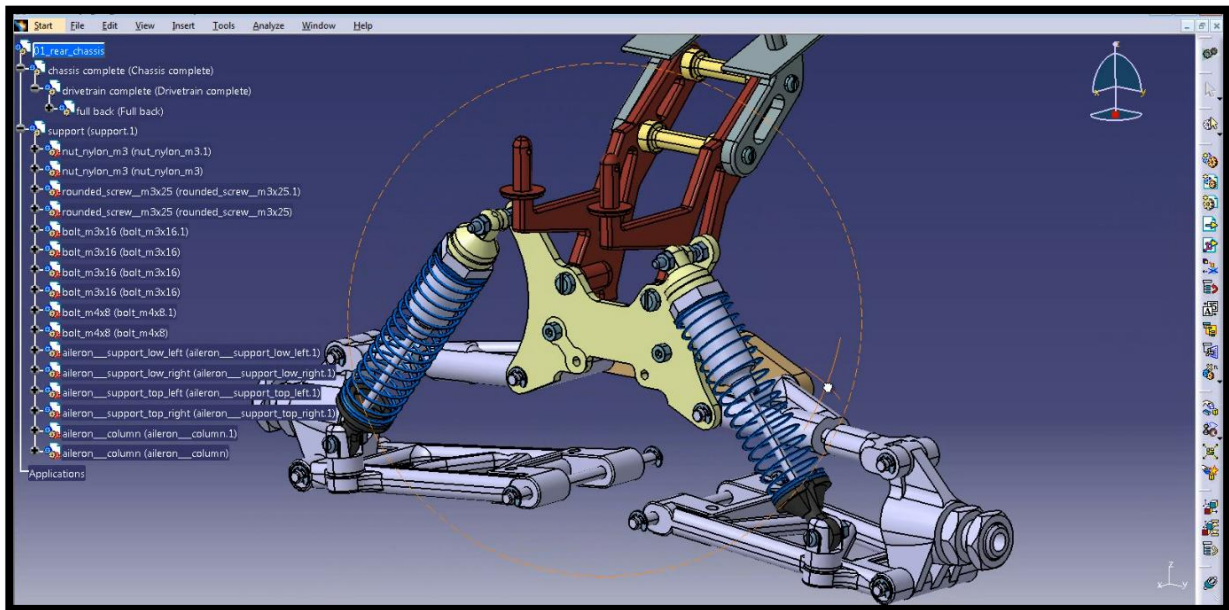


Figura 4. 3 Interfaz CATIA

#### 4.2.4 Inventor

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 modelo en 2D como para el 3D. Se logra realizar un proyecto en todas sus fases, boceto de la pieza, ensamble, animación, plano de componentes, materiales utilizados.

La animación del modelo 3D, se realiza en el entorno de “Inventor Studio”, el movimiento 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. Además, 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.



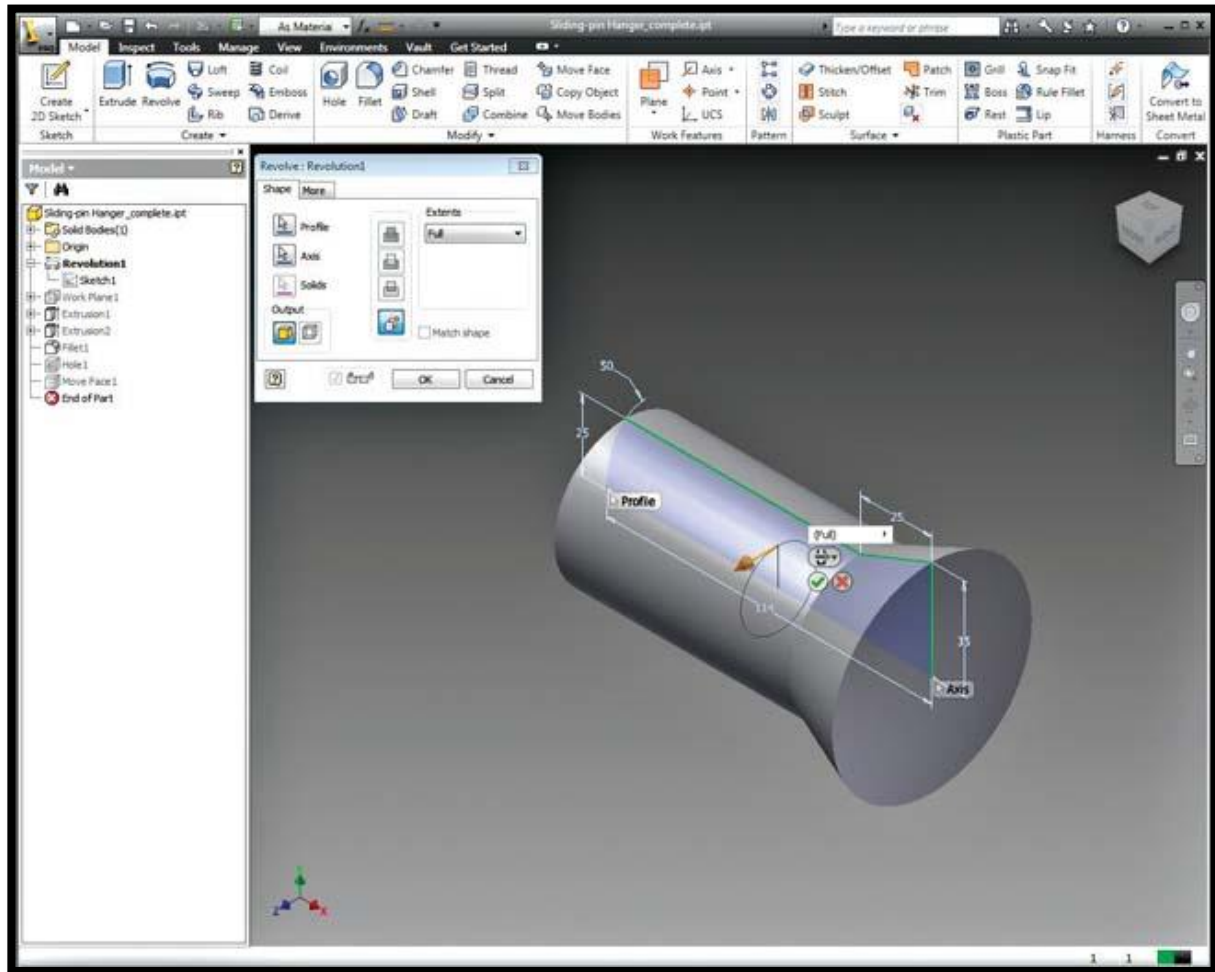


Figura 4.4 Interfaz Inventor

### 4.3 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 respecta a ese tema todos son muy eficaces.

Otra cuestión motivo de análisis, fue el entorno de simulación. En este caso el primer programa descartado fue VariCad, este software no incluye en su paquete la simulación de sus proyectos.

De SolidWorks y CATIA, son programas de empresas muy reconocidas con años de experiencia en el desarrollo en materia de ingeniería, softwares demasiado potentes y de un dinamismo muy eficaz. Para la envergadura de este proyecto no son los más adecuados.

Finalmente el software escogido es Inventor. Por comodidad, la interfaz de usuario es muy fácil de manejar, cumplen con las características de ser un software de modelado paramétrico, realiza perfectamente modelos en 3D, simulaciones, documentación referida a las piezas de cada ensamble, gran variedad de tipos de materiales y despiece de componentes.

Otro punto muy importante que se consideró para la elección, fue facilidad para adquirir el programa, la Universidad de la Laguna cuenta con un convenio con Autodesk que permite a los estudiantes y profesorado disponer del software en formato educacional, tanto en ordenadores personales como las aulas universitarias.

#### **4.4 Etapas para realizar la simulación**

Teniendo en cuenta la teoría de varios autores sobre los pasos necesarios para realizar un estudio de simulación [10], se llegó a la conclusión que las etapas más relevantes útiles para este proyecto fueron:

##### **Definición del sistema**

El sistema real simulado fue la estación de acarreo de piezas, el objetivo era simular el movimiento del brazo de transferencia, que transporta piezas de la estación de procesado a la estación clasificación de piezas.

##### **Variables a concluir en el modelo**

Las variables fueron de tipo cuantitativo, específicamente los tiempos en que se ejecutaba el transporte de las piezas defectuosas y no defectuosas de una estación a otra.

##### **Recolección y análisis de los datos del sistema**

La recolección de datos se realizó en el laboratorio de automatización, se efectuó la medida milimétrica de cada pieza que componía la estación y se tomaron fotos de su aspecto para lograr un modelo lo más semejante a sistema real.

##### **Programación del modelo**

Se realizó un estudio para la elección del software adecuado. En él, se analizaron varios programas y sus características, finalmente se escogió el más acorde a las necesidades del proyecto.

### **Realización y validación del modelo**

Obtenido modelo se verificó la similitud con el sistema real. En esta etapa se comprobó los tiempos de ejecución del transporte de las piezas, se realizó videos de la simulación y se comparó con los videos reales de la estación. Por último se corrigieron los errores detectados.

### **Análisis y crítica de los resultados**

Se contrastó el grado de ajuste entre el modelo virtual y el real con los parámetros previamente establecidos.

## **4.5 Ventajas de la simulación**

- El uso de la simulación como método de aprendizaje logrará que los alumnos abarquen una práctica con más seguridad, es un método didáctico, donde el alumno se divierte y aprende un tema determinado.
- Si previamente ha utilizado la simulación como método de aprendizaje, ganará agilidad en la asignatura en cuestión, logrando reducir el tiempo que tarda en realizar una práctica.
- Las grandes empresas actualmente utilizan el diseño y definición de un modelo en 3D, para analizar y evaluar un producto antes de ser fabricado, lo genera una reducción de tiempo y costo. Antiguamente era imposible de evitar estos costos con los métodos de prueba y error realizados para la verificación del producto.
- El trabajar en el diseño de un modelo virtual, no implica interrumpir las operaciones cotidianas de las empresas.
- La simulación permite el análisis de problemas complejos para los que no están disponibles resultados analíticos. La mayoría de los problemas de mundo real encajan en esta categoría. La simulación proporciona una alternativa práctica.
- La simulación no solo analiza proyectos nuevos, sino que además puede mejorar sistemas reales existentes, logrando un mejor rendimiento.

## Capítulo 5. Inventor

### 5.1 ¿Qué es inventor?

Creado especialmente para desarrollar prototipos digitales, útiles para verificar el estado y la eficacia de un producto antes de fabricarlo. Es un programa de diseño mecánico industrial, que cuenta con un conjunto completo de herramientas para producir, simular y documentar modelos en 3D. Con este programa, los ingenieros pueden integrar los dibujos 2D de AutoCAD y los datos 3D en un modelo digital único, para crear una representación virtual del producto final. Para los empresarios y fabricantes representa una ventaja en todos los sentidos, ya que reducen costos, optimizan el tiempo de diseño y fabricación, por último logran productos muy fiables para los mercados.

Inventor un CAD 3D, es un software que proporcionan funcionalidad avanzada para diseño de ingeniería mecánica, análisis de elementos finitos, simulación de movimiento, gestión de datos, sistemas enrutados y diseño de moldes, además de soluciones que aumentan la productividad de CAD.

### 5.2 Características de Inventor

La versión 2015 de este software que ha producido la empresa Autodesk, dispone de mejores características de productividad, de gran importancia para comunicar los diseños y compartir los modelos de forma amplia. Con entornos de trabajo muy intuitivos, simulación de movimiento y análisis de tensión integrados.

Se exponen las características más importantes de este software:

- **Integración con AutoCAD y compatibilidad con DWG**

Herramienta para utilizar documentación disponible en otros formatos, permite que el usuario se enfoque en el diseño y creación de un mejor prototipo digital, evitando operaciones engorrosas de transcripción de archivos. Facilita la operación de compartir datos de diseño estratégicos con colaboradores y clientes.

- **Compatibilidad con BIM.**

BIM (*Building Information Modeling*) junto con la funcionalidad del Exchange permite el intercambio de datos entre el software Inventor y los productos Autodesk,

como por ejemplo Revit y AutoCAD Architecture. Esta es una herramienta muy útil para los fabricantes e ingenieros mecánicos que trabajan con arquitectos, constructores y contratistas.

- **Gestión de datos**

Aplicación de gestión de datos Vault integrada. Permite a los equipos de trabajo ya sea de diseño o ingeniería organizar, administrar, realizar seguimiento de un proyecto de una manera rápida y segura.

- **Documentación de diseño y fabricación.**

Entre las características de documentación de diseño y fabricación se incluyen: compatibilidad con las principales normas de dibujo, lectura y escritura directas del formato DWG auténtico, vistas de plano automáticas, actualizaciones de plano automáticas, lista de materiales, lista de piezas asociativas. Mediante esta herramienta el programa genera documentación de diseño de ingeniería y fabricación a partir de los modelos, lo que ayuda a reducir errores y a terminar los diseños en menos tiempo.

- **Diseño de ensamblajes.**

Utilizando las restricciones mecánicas y manejando los grados de libertad de cada pieza, se obtiene el ensamble deseado. Inventor proporciona un conjunto de herramientas para controlar y administrar los datos generados por los diseños de ensamblajes grandes y complejos.

- **Diseño variado de piezas específicas**

El programa ofrece el diseño de piezas específicas, como por ejemplo cables y mazos, muy útiles ya que la mayoría de los productos o máquinas incluyen sistemas eléctricos que exigen un diseño meticuloso de estas piezas. Igual de importante es el diseño tubos y tuberías, para ello el software simplifica y agiliza el diseño de elementos complejos en esta categoría, entre ellos encontramos tubos, tuberías y mangueras flexibles.

- **Diseño de moldes y mecanizado.**

Característica innovadora en la versión de inventor 2015. Adelanta facetas clave del diseño de moldes de inyección para piezas de plástico. Con esta aplicación se realiza el diseño de moldes completos. El programa facilita el proceso de diseño de moldes con un trabajo intuitivo que resulta natural para los usuarios. Se incluye también la simulación del molde y la validación con las herramientas de análisis del flujo de plástico. Se puede obtener la velocidad de flujo del material, evaluar la contracción y definir los parámetros del proceso.

- **Análisis de elementos finitos integrado.**

Mayor exactitud y fiabilidad de los análisis de tensión de los diseños.

- **Simulación de movimiento integrada.**

El software cuenta con gran variedad de técnicas para la simulación, en Inventor Studio, se logra no sólo la animación de piezas sino que también renderizar las animaciones para dar un acabado más realista al modelo.

## 5.3 Entorno de trabajo

El paquete diseño Autodesk Inventor se compone de cinco módulos o entornos, principalmente usados para desarrollar tareas específicas dentro de la labor de diseño de piezas en un modelo digital.

Para la realización de la simulación de la Estación de acarreo de piezas de la planta Festo se utilizaron estos entornos de trabajo. A continuación se explica cada módulo y la manera de empleo.

### 5.3.1 Interfaz de Inventor

El software inventor es muy intuitivo y fácil de manejar. En la figura 5.1 se aprecia la interfaz que aparece cuando se abre el programa. En esta parte el usuario indica lo que desea realizar, puede ser empezar un nuevo proyecto, abrir una pieza, ensamble o documentación adicional.

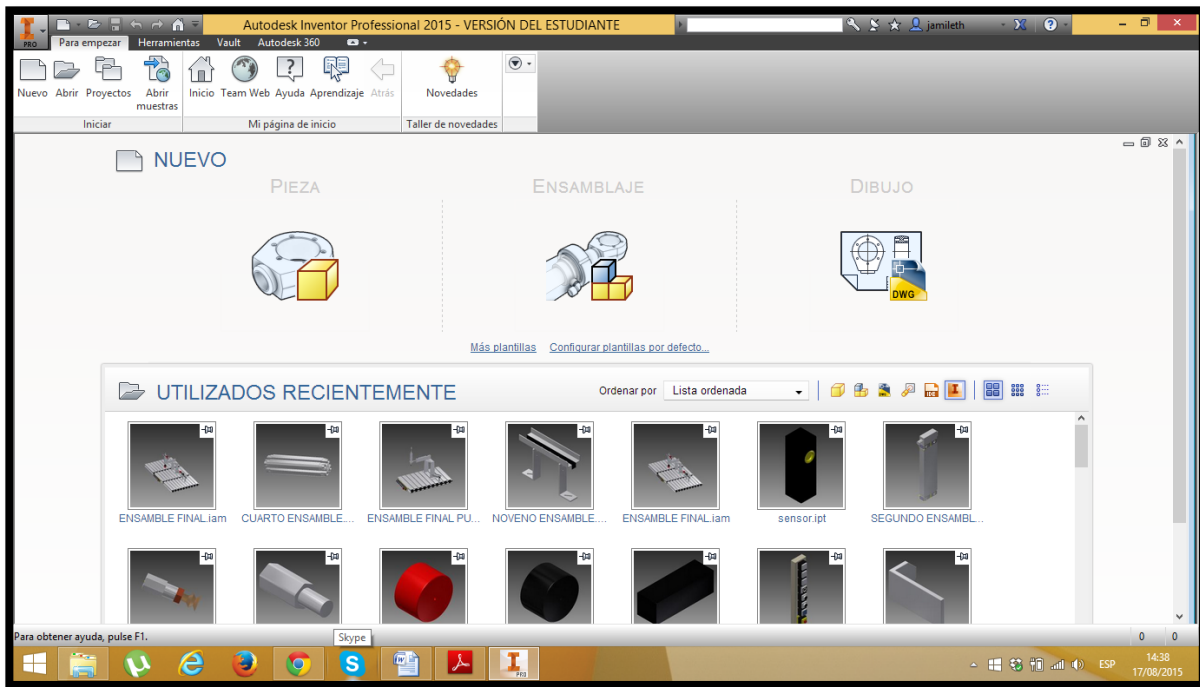


Figura 5. 1 Interfaz Inventor

Cuando el usuario indica lo que va realizar aparece la interfaz de trabajo, en la figura 5.2 se destaca los elementos más importantes de esta:

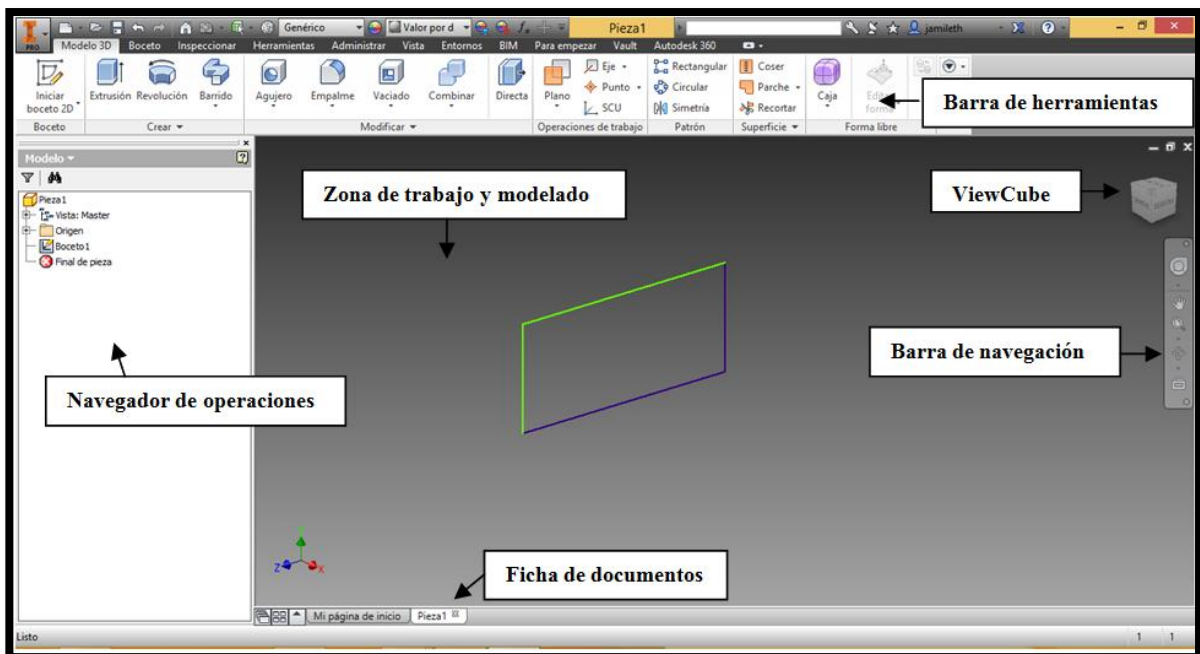


Figura 5 2 Interfaz de trabajo Inventor

### 5.3.2 Entorno de operaciones

El entorno de operaciones permite la creación de modelos en tres dimensiones. Primero se realiza la geometría en dos dimensiones o boceto de la pieza, después para obtener el sólido, se aplica de acuerdo a la necesidad, operaciones como revoluciones, extrusiones, solevaciones, etc.

A continuación se muestra como se creó una de ventosa (figura 5.3), parte del bloque de piezas de la Estación de acarreo de piezas

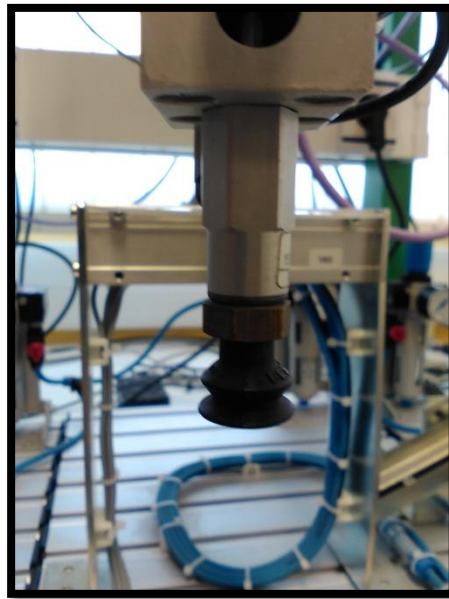


Figura 5.3 Ventosa

Cuando se crea un documento nuevo, el programa abre una ventana donde se indica la extensión del archivo y la unidad de medida del modelo, en este caso el archivo es una pieza, lo que corresponde una extensión .ipt (*Inventor Parts*) y las unidades de medidas en milímetros (Figura 5.4)



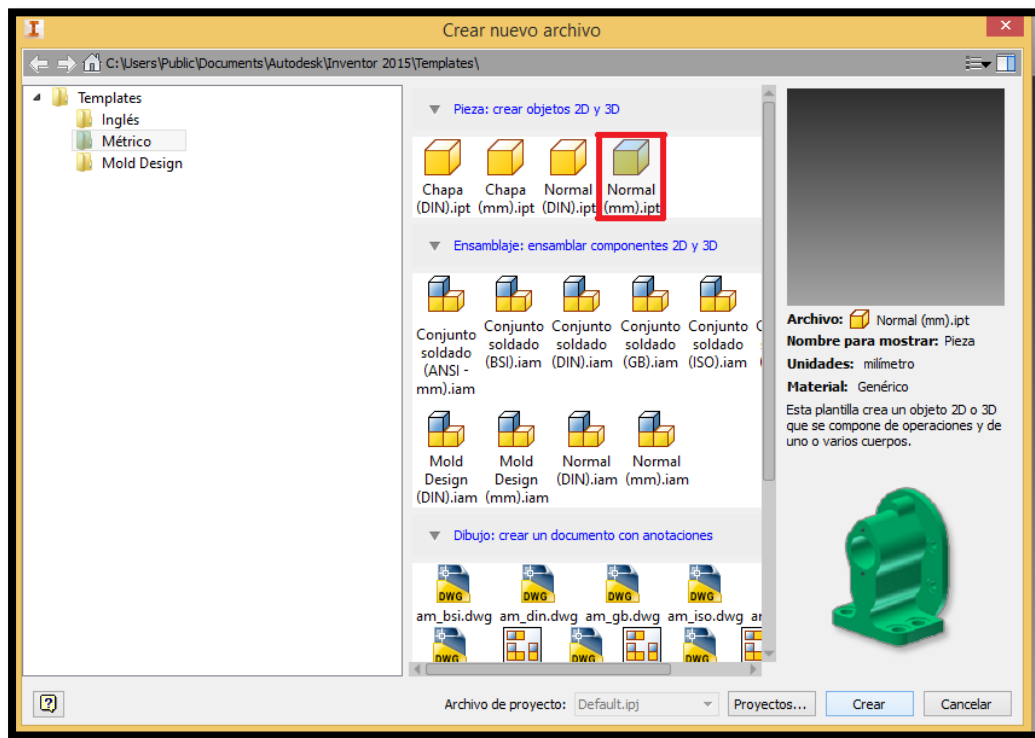


Figura 5.4 Formato de archivo

Para realizar el boceto de la pieza se accede al entorno “Boceto 2D”. Para la ventosa se dibujaron seis círculos cada uno de ellos separados a una distancia específica, fue posible haciendo uso del menú de herramientas, con las opciones círculo y cota.

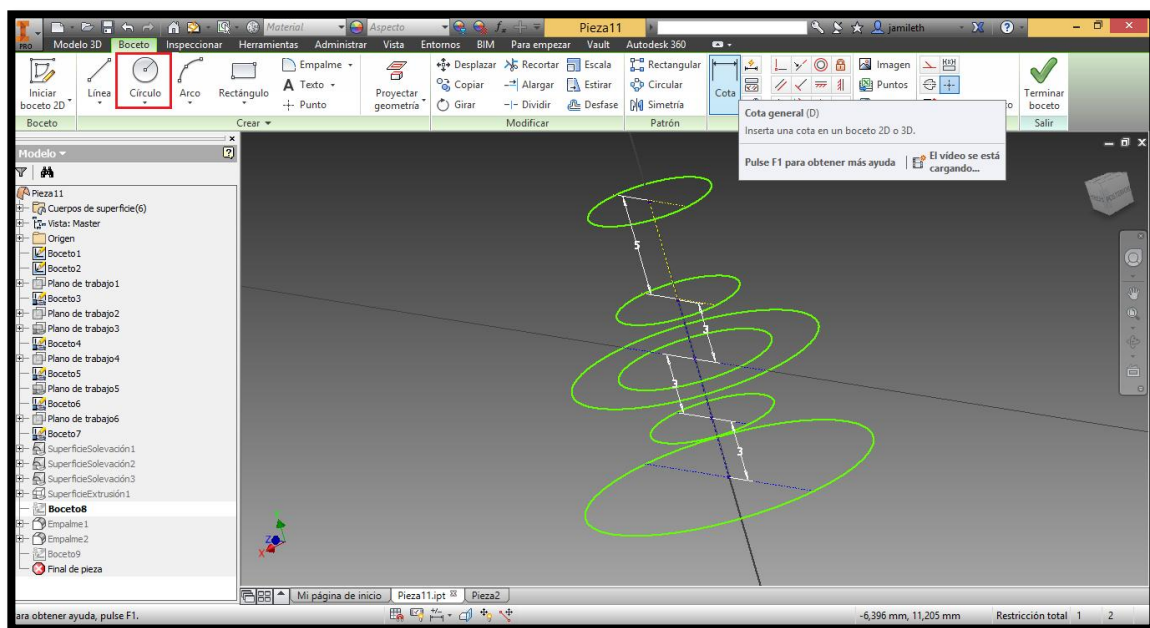


Figura 5.5 Boceto de la ventosa

La dificultad de la pieza, consistía en que los círculos estaban en un mismo plano pero a un desfase diferente, para solucionarlo se empleó de la barra de herramientas, la opción plano, donde utilizando el plano de origen XZ, se logró desfazarlo las veces que fuese necesario (Figura 5.6).

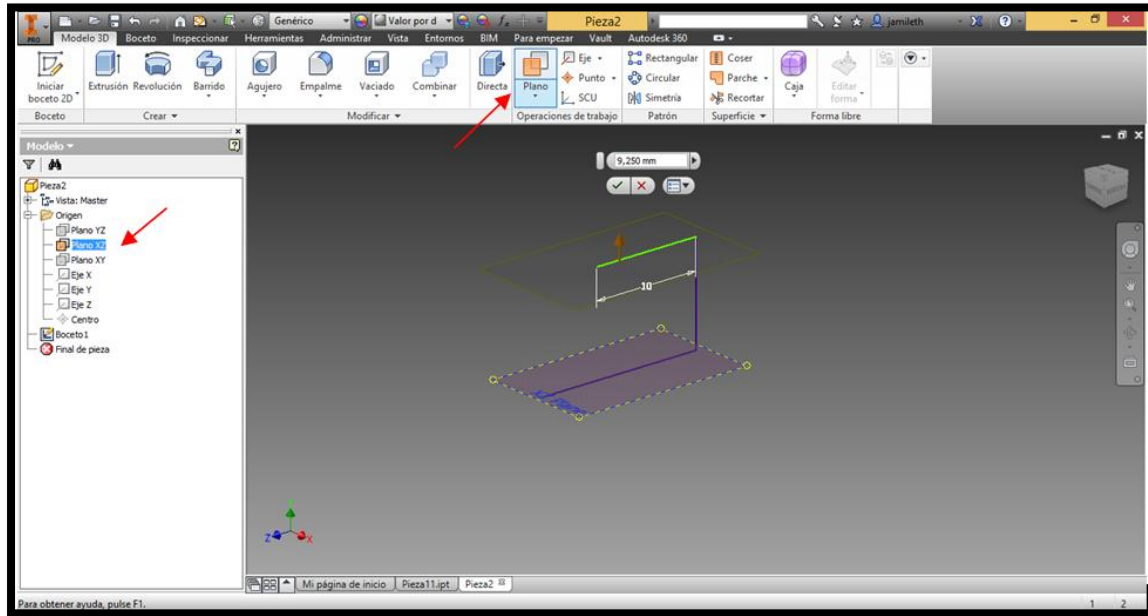


Figura 5.6 desfase de planos

Cuando se ha terminado el boceto, se procede a dar solidez a la pieza. Para la ventosa se utilizó de la barra de herramientas de modelado, las opciones extrusión y solevación (Figura 5.7 y Figura 5.8)

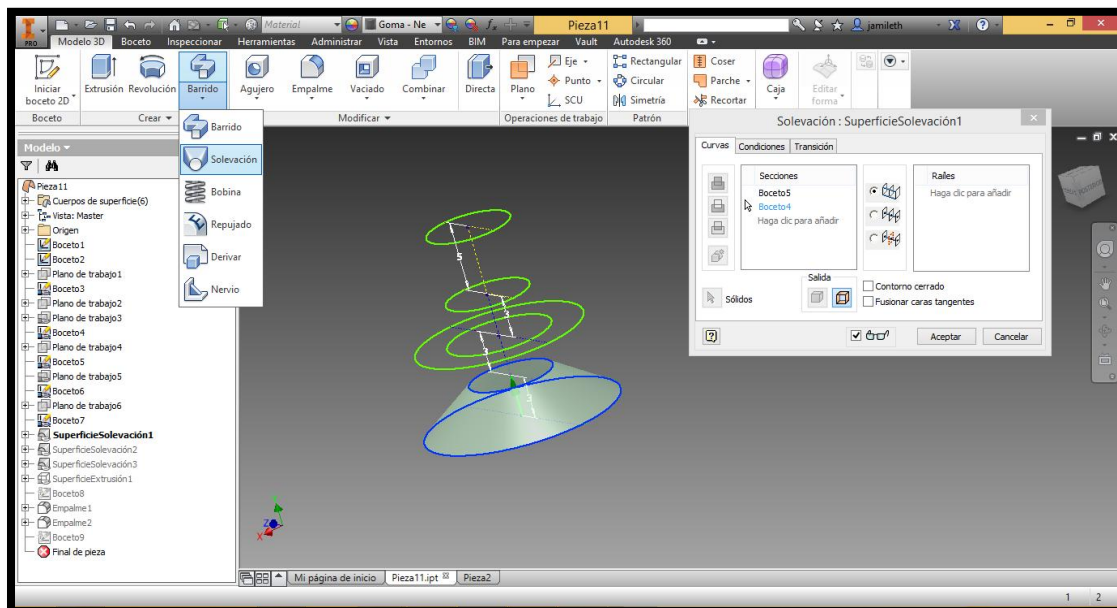


Figura 5.7 Solevación de pieza

El programa permite en todo momento efectuar correcciones en el diseño de las piezas, evitando empezar un modelo desde el inicio.

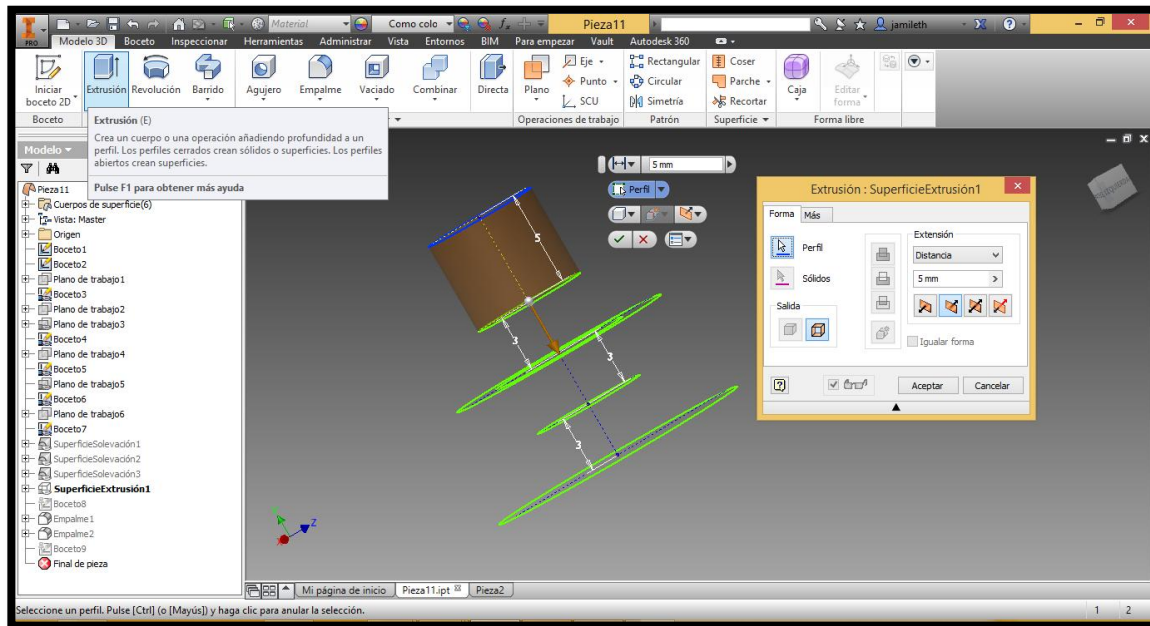


Figura 5.8 Extrusión de la pieza

El siguiente paso es conseguir que la pieza tenga un acabado lo más parecido a la pieza real, para ello se moldean las uniones entre cada círculo mediante la opción empalme.

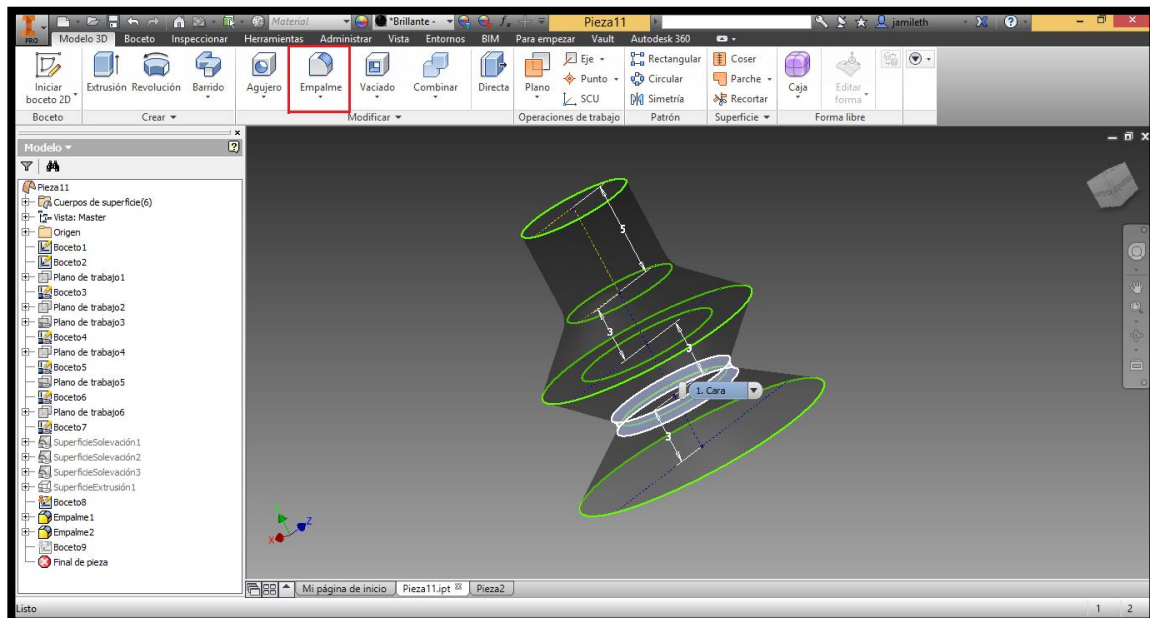


Figura 5.9 Empalme de la pieza

Finalmente se ocultan los bocetos, se escoge el material de la pieza, aspecto a destacar ya que el Software cuenta con gran diversidad de materiales y apariencias que logran un realismo total a la pieza. En la figura 5.10 se observa el resultado final de la ventosa.

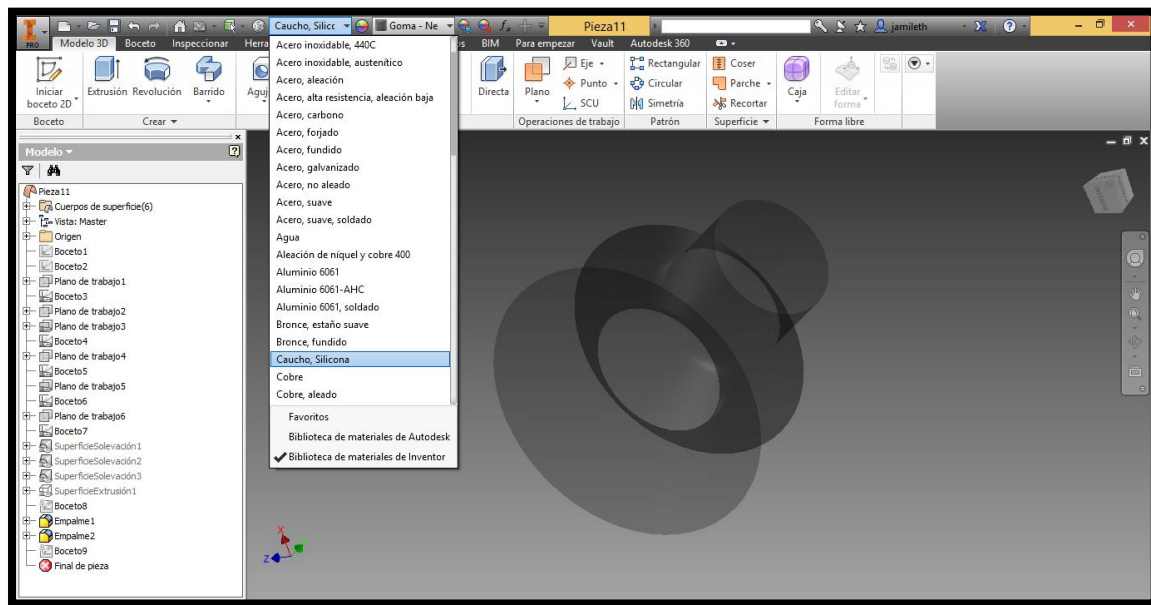



Figura 5. 10 Pieza terminada


### 5.3.3 Entorno de ensamblaje


En este módulo se establece la unión del conjunto de piezas realizadas en el entorno anterior. Para realizar el ensamblaje es de suma importancia, el uso adecuado de las restricciones mecánicas entre piezas, estas definirán la posición y movilidad del modelo completo, teniendo en cuenta los grados de libertad de cada elemento. Las restricciones pueden variar de un modelo digital a otro, posteriormente se describen las restricciones [11] utilizadas en este proyecto:



Figura 5. 11 Restricciones de ensamblaje

 Restricción de coincidencia coloca los componentes frente a frente o uno junto al otro con las caras niveladas. Suprime un grado de traslación lineal y dos grados de rotación angular entre las superficies planas.

 Restricción angular: permite colocar las aristas o las caras planas en dos componentes a un ángulo especificado para definir un punto de giro. Suprime un grado de libertad en la rotación o dos grados de rotación angular entre las superficies planas.

 Restricción de inserción: combinación de una restricción de coincidencia cara a cara entre caras planas y una restricción de coincidencia entre los ejes de los dos componentes. Tiene un grado de libertad de rotación

El programa Inventor para mejorar el rendimiento de la memoria utilizada, tiene la opción de controlar las restricciones y prescindir de las innecesarias, esta opción se activa en el menú Archivo/opciones/ensamblaje (Figura 5.12).

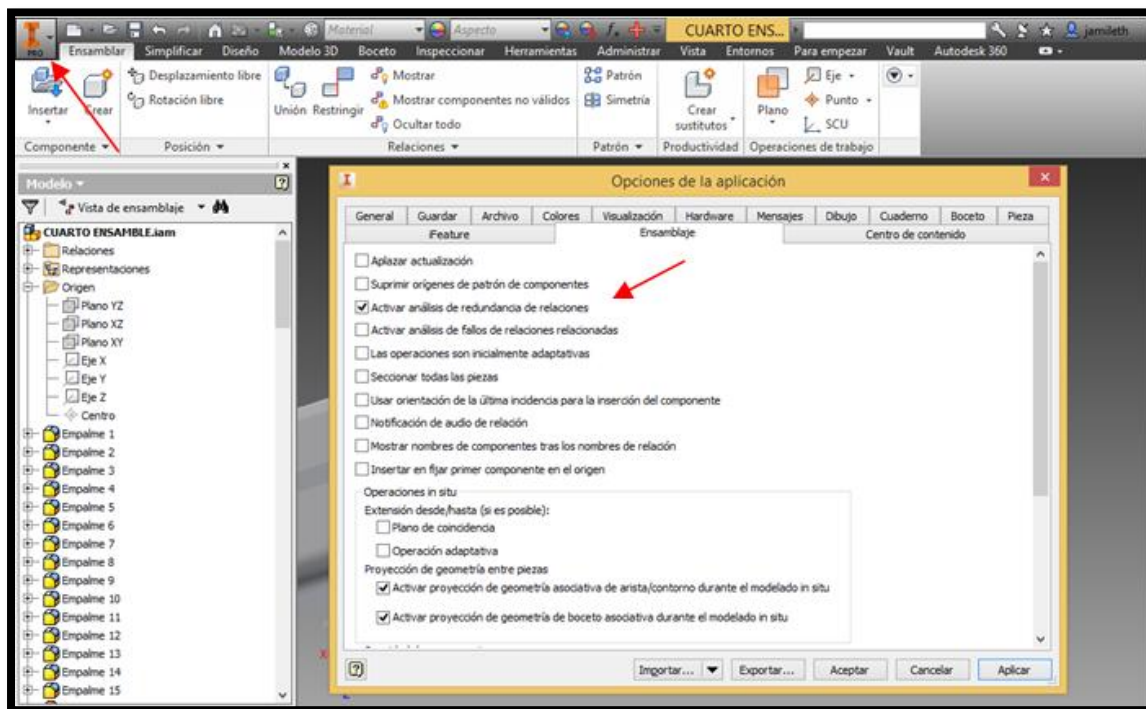



Figura 5.12 Análisis de restricciones

 El símbolo indica restricción redundante, el icono se localiza a la izquierda de la restricción afectada. Para eliminar la restricción se pulsa el botón derecho del ratón y se selecciona la opción suprimir (Figura 5.13).

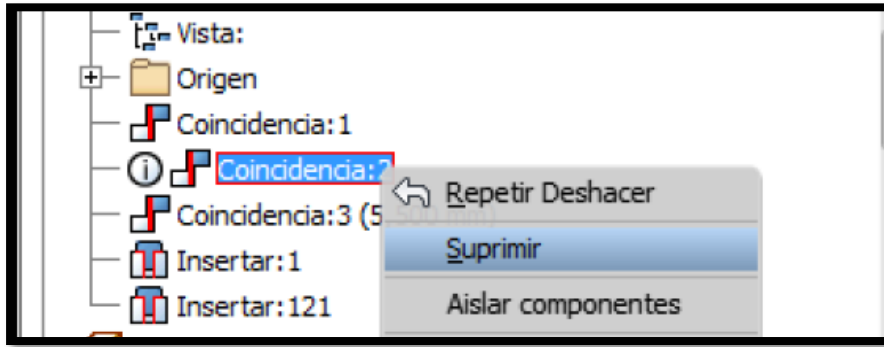


Figura 5.13 Restricción redundante

⚠ El símbolo indica existencia de error en la restricción, el símbolo aparece en lado izquierdo de la restricción. Para solucionarlo se pulsa el botón derecho del ratón y se selecciona recuperar para desactivar o suprimir (Figura 5.14).

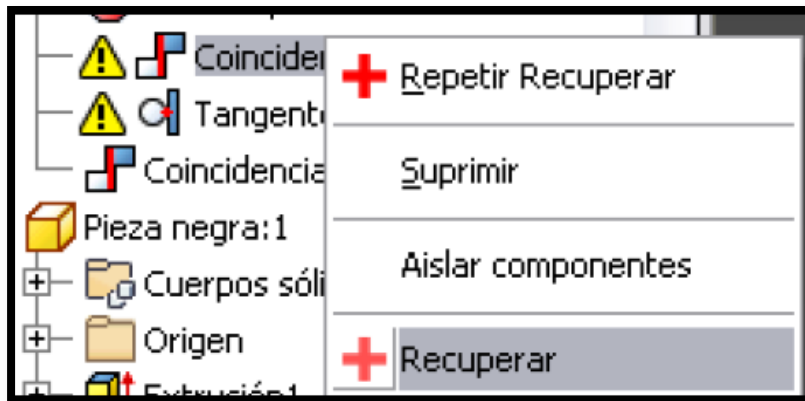


Figura 5.14 Error en la restricción

Para entender mejor el módulo de ensamblaje, se explicara cómo se realizó uno de los ensamblajes de la Estación de acarreo de piezas, compuesto por piezas individuales y por subensamblajes hechos previamente.

En primer lugar se escoge un nuevo documento, en la ventana siguiente se elige la opción conjunto solado, el archivo tiene un formato .iam (*Inventor Assembly*), en la figura 5.15 se observa dicha ventana.

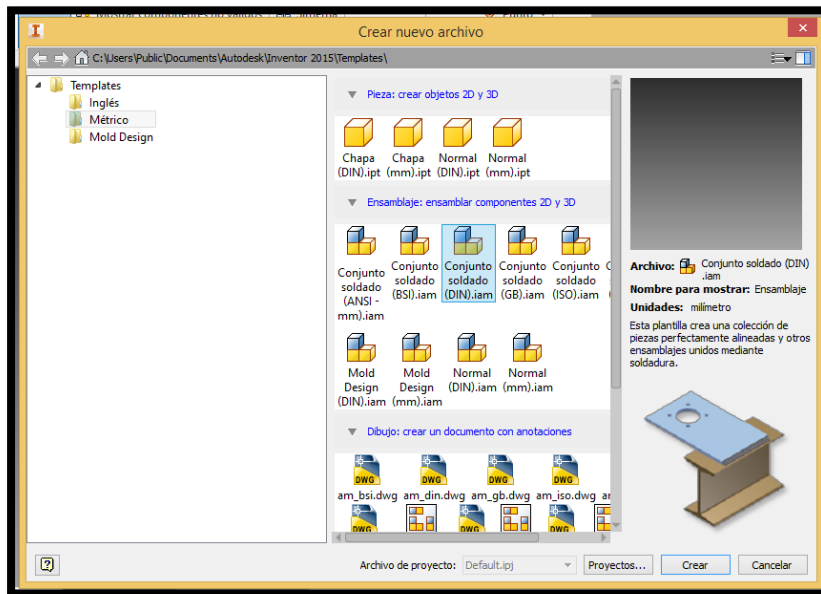


Figura 5.15 Elección de conjunto soldado

Una vez en la interfaz de trabajo, para insertar el conjunto de piezas, se utiliza la opción insertar (Figura 5.16)

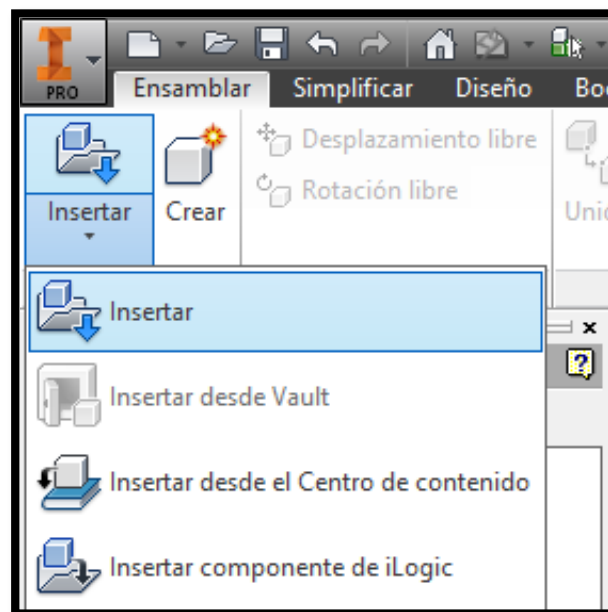


Figura 5.16 Insertar piezas

Luego se abre una ventana donde se debe indicar la ubicación de la carpeta que contiene todas las piezas y seleccionar las necesarias, en este caso se escogió la mesa y el primer subensamblaje (Figura 5.17)

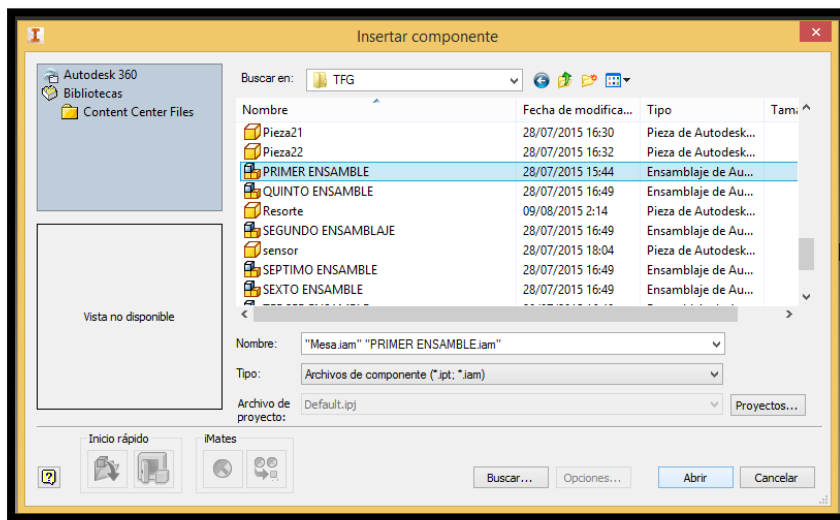


Figura 5.17 Elección de piezas

Para facilitar el trabajo se fija un elemento. En la Estación de acarreo de piezas se fijará la mesa, evitando el movimiento de todo el conjunto (Figura 5.18).

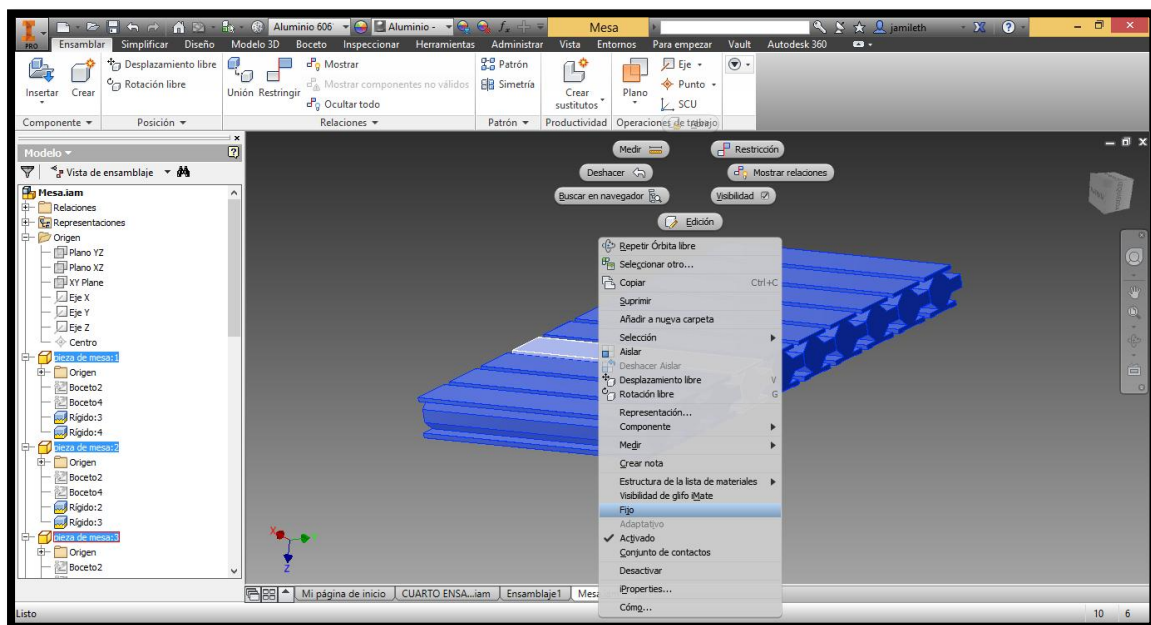


Figura 5.18 Elemento fijo

El objetivo de la primera restricción es colocar la armadura encima de la mesa, la cara inferior de las patas de la armadura debe de coincidir con la cara superior de la mesa, ambas serán perpendiculares entre sí (Figura 5.19).



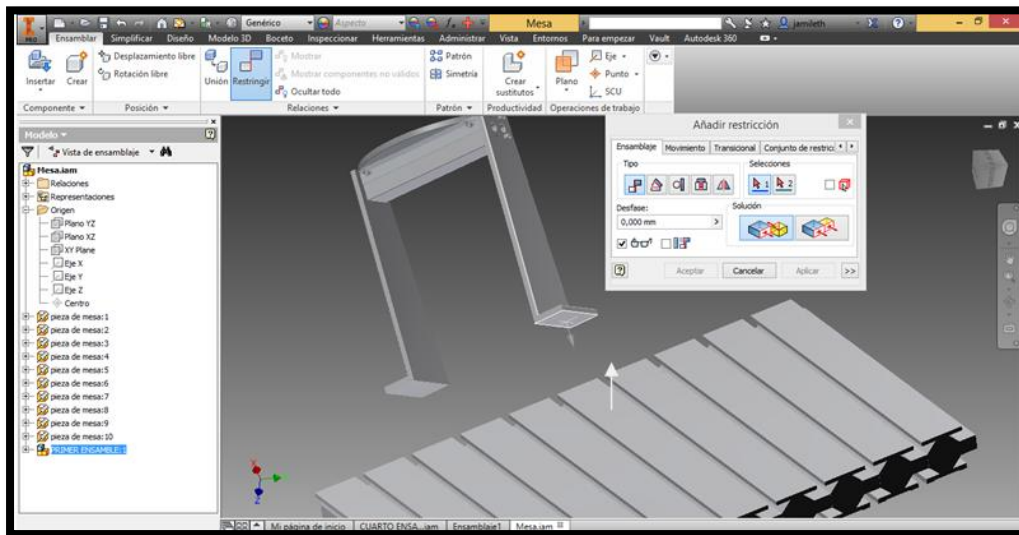


Figura 5.19 Restricción de coincidencia

En la figura 5.20 se observa el resultado.

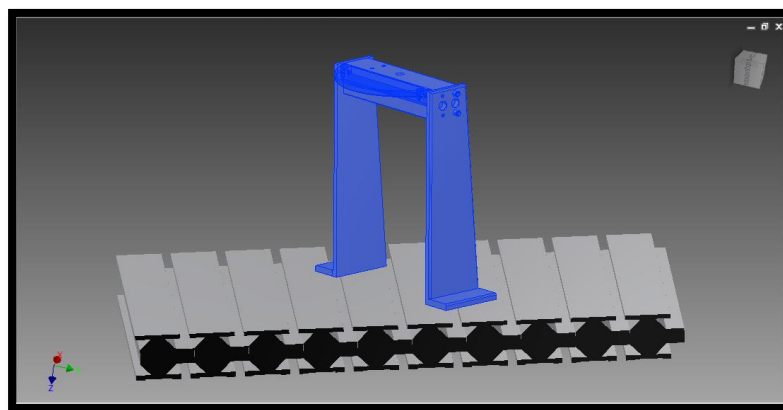


Figura 5.20 Armadura sobre mesa

Se aplica una segunda restricción de coincidencia para lograr que la armadura quede paralela a las líneas de la mesa (Figura 5.21).

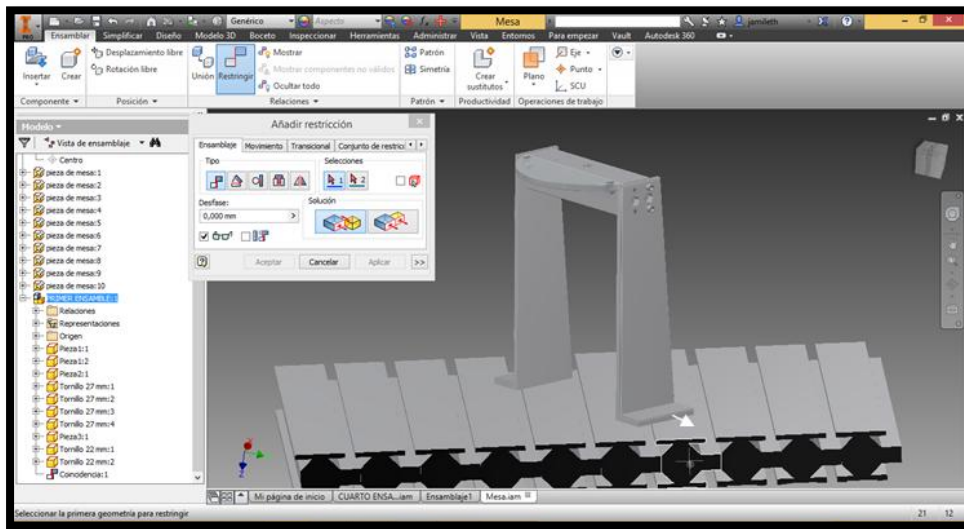


Figura 5. 21 Segunda restricción

Para ubicar la armadura donde el usuario crea conveniente, el programa permite variar la distancia entre las caras que existe la coincidencia (Figura 5.22).

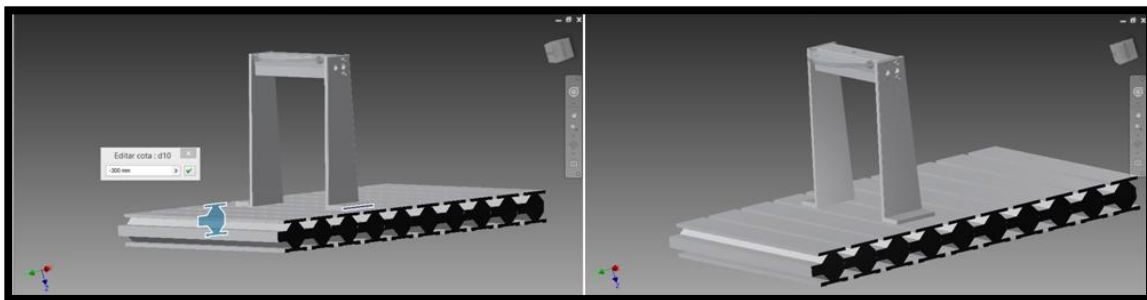


Figura 5.22 Armadura en posición correcta

Otra restricción muy utilizada durante en el ensamblaje fue la restricción de inserción, debido a que la Estación de acarreo de piezas cuenta con muchos tornillos, además de dos cilindros de extensión vertical y horizontal, por esta razón cabe destacarla. En la figura 5.23 se observa como mediante la restricción de inserción se logra introducir el perno en el hueco de la armadura.

En ella existe una coincidencia entre la cara inferior del perno y la cara superior del agujero, también debe de haber una coincidencia de ejes, el brazo queda con la libertad de girar sobre un solo eje.

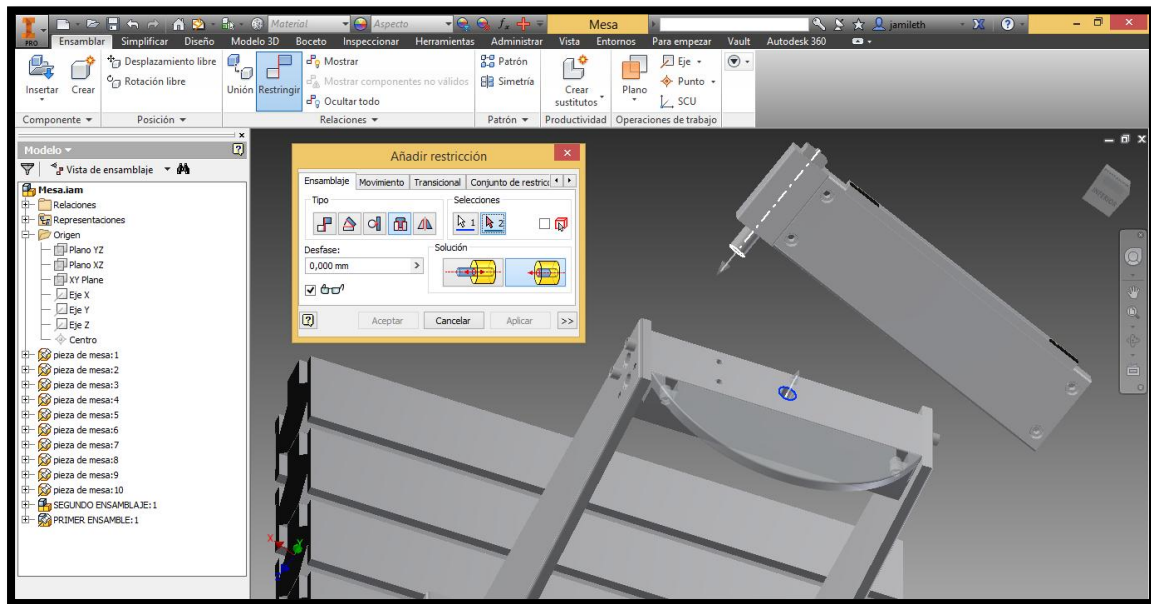


Figura 5.23 Restricción inserción

De forma similar, es decir haciendo uso de las restricciones, se realiza este procedimiento con todas las piezas y subensamblajes hasta obtener por completo la Estación de acarreo de piezas. En la figura siguiente se aprecia el resultado final.

Gracias a que Inventor es un software de modelado paramétrico, cualquier cambio o corrección en una pieza, se verá reflejado en el ensamblaje final.

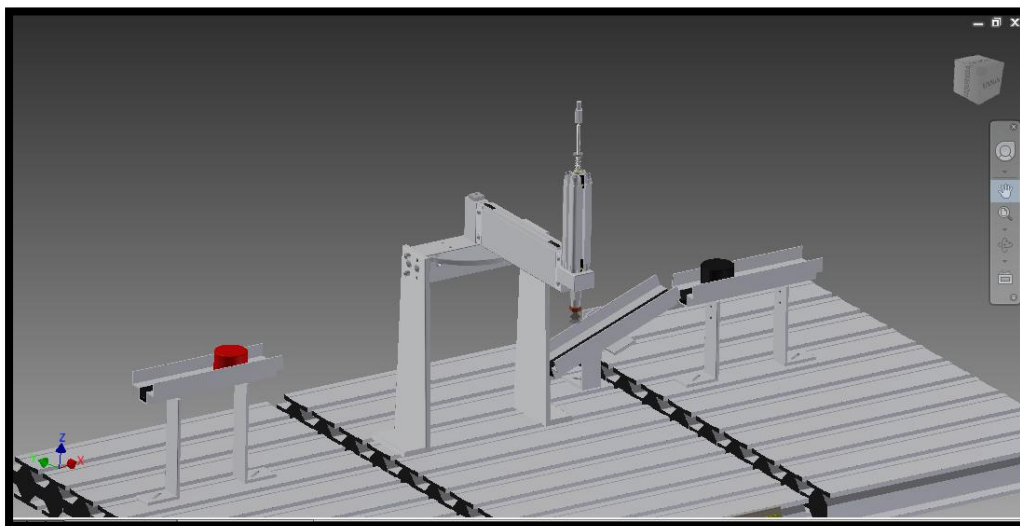


Figura 5.24 Ensamble completo

### 5.3.4 Entorno de simulación

En este módulo el usuario puede realizar gran variedad de trabajos relacionados con la animación del prototipo digital. En él se crean las animaciones, teniendo en cuenta las restricciones y grados de libertad de cada parte del ensamble, se realiza videos de la animaciones, el video puede contener diferentes vistas, esto se logra mediante el uso de cámaras, se renderiza el video para obtener una mejor calidad de imagen, etc.

Para acceder a este entorno, se hace desde la interfaz del programa, seleccionando del menú, la pestaña entornos/ Inventor Studio (Figura 5.25).

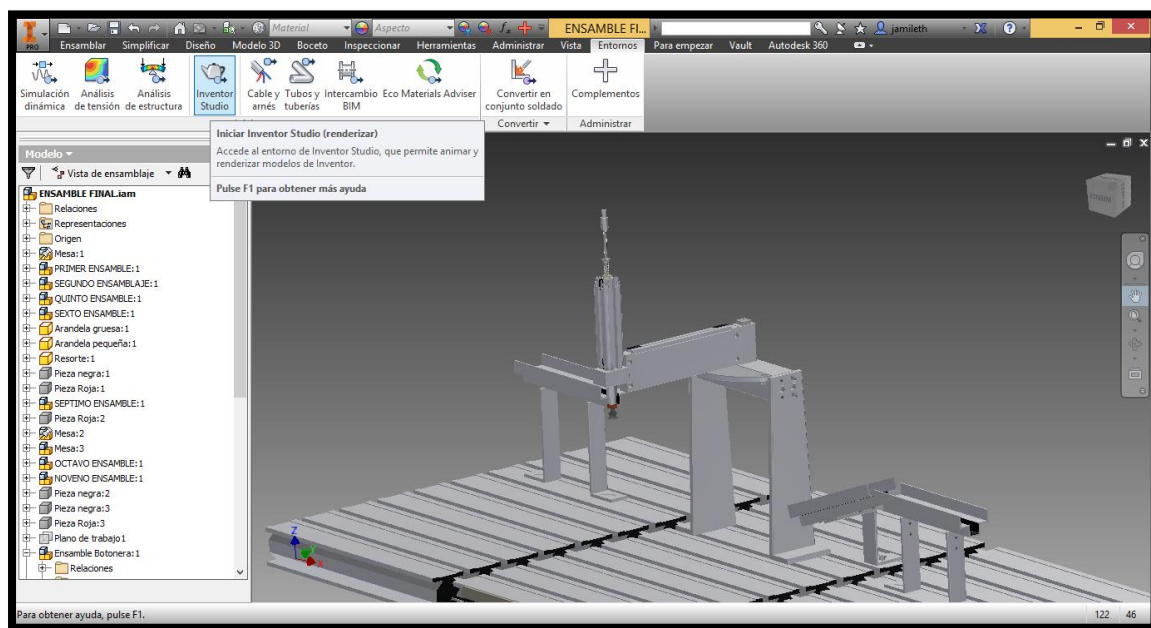


Figura 5.25 Acceso a Inventor Studio

Dentro de modulo se aprecia la barra de herramientas con todas sus opciones (Figura 5.25).

Durante este proyecto en varias ocasiones, se ha mencionado lo intuitivo que es el software Inventor, pero es importante que el usuario interactué con el entorno, investigue y descubra la infinidad de herramientas que posee este programa, para lograr un avance más rápido en los conocimientos y en la técnica.

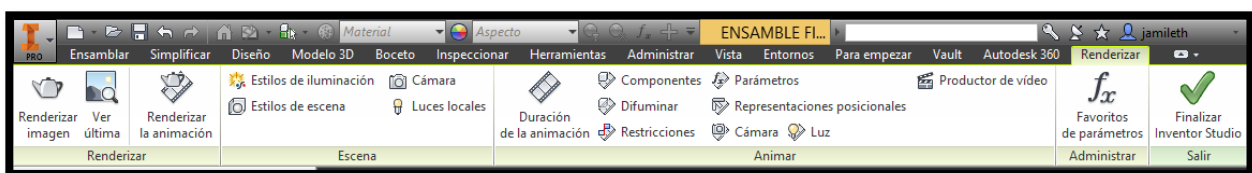


Figura 5.26 Herramientas Inventor Studio

La animación se construye convirtiendo las restricciones mecánicas del ensamblaje en uniones estándar. Con ellas se podrá definir la posición inicial y final de la trayectoria a simular.

Para crear una nueva animación, se selecciona del navegador de operaciones la opción animación/nueva animación (Figura 5.27).

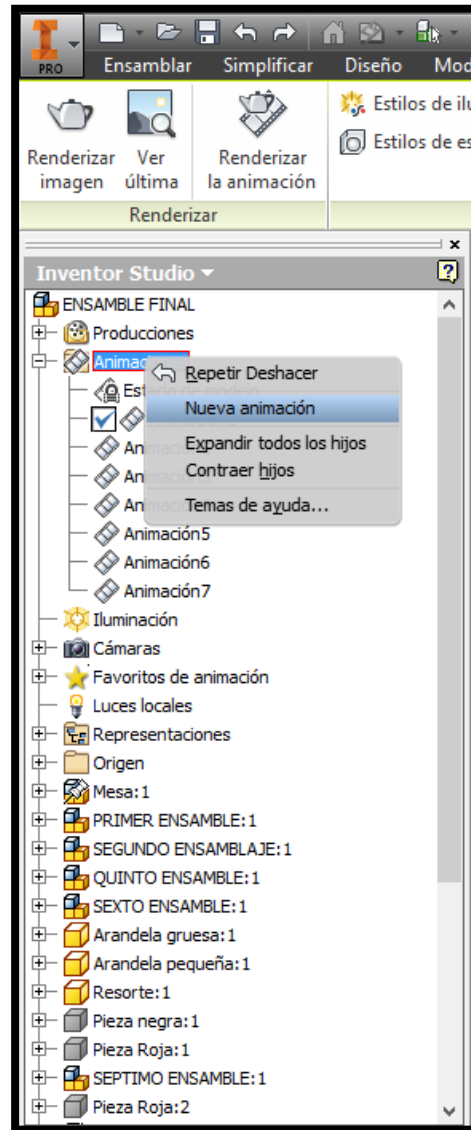


Figura 5.27 Creación de nueva animación

En esta etapa el usuario debe indicar que restricción animar, para esto se dirige a la restricción en cuestión y selecciona animar restricción (Figura 5.28)

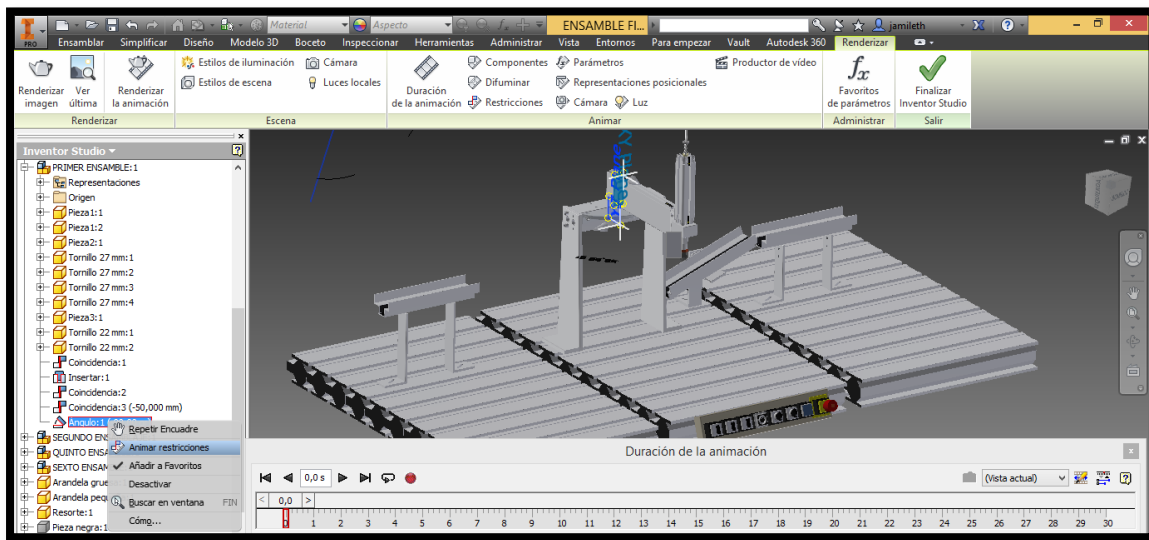


Figura 5.28 Animar restricción

Después aparecerá una ventana (Figura 5.29), que indica el grado de libertad que tiene la restricción. En este caso es una restricción angular, el inicio de posición es 90 grados, final de posición 0 grados y el tiempo de ejecución de esta acción se realizara en cuatro segundos, desde el minuto cero al minuto cuatro.

El usuario tiene la libertad de realizar la animación a su conveniencia, en el caso preciso de la Estación de acarreo de piezas, los tiempos utilizados fueron lo más parecidos al video real de la estación.

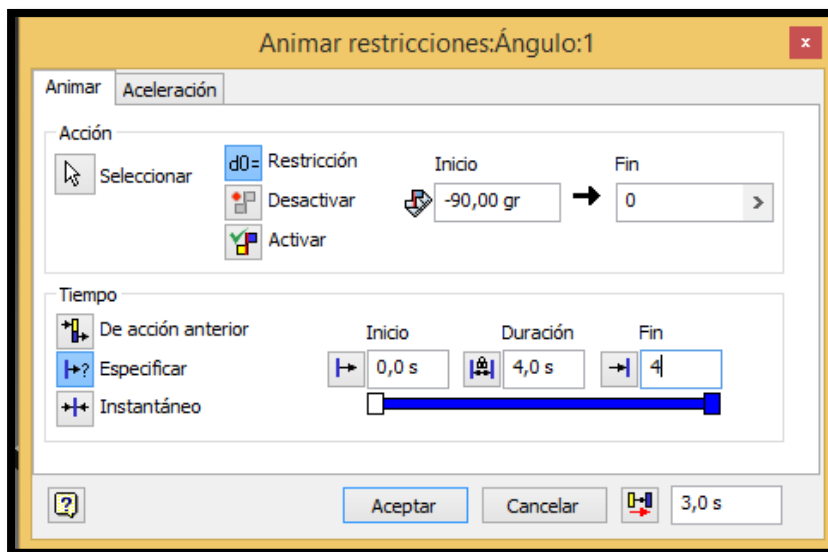


Figura 5.29 Ventana para configurar animación de restricción

Cuando se realiza la animación a gusto del usuario, el siguiente paso es agregar a dicha animación cámaras para tener una perspectiva desde puntos diferentes.

Para ello el usuario debe situarse en el navegador de operaciones seleccionar la opción cámara/crear cámara según vista (Figura 5.30).

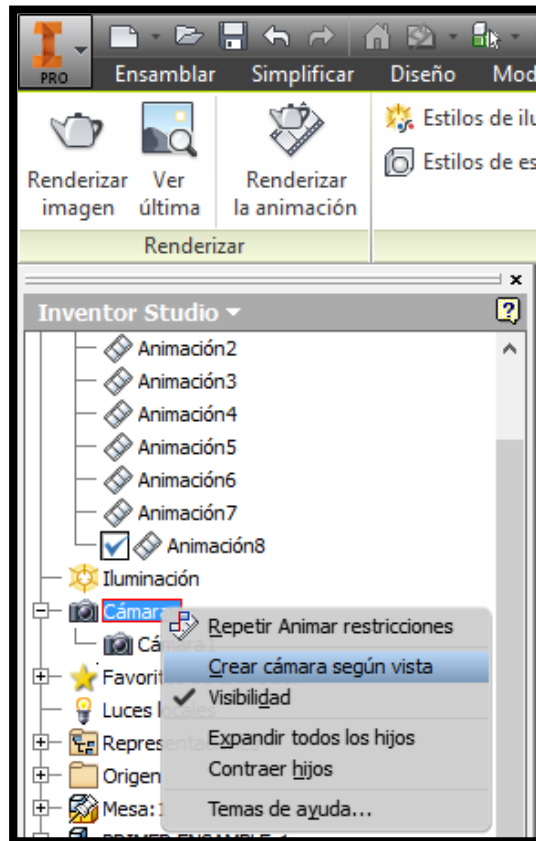


Figura 5.30 Creación de cámara

Para seleccionar la cámara, se debe ir al panel indicador de la animación. (Figura 5.31).

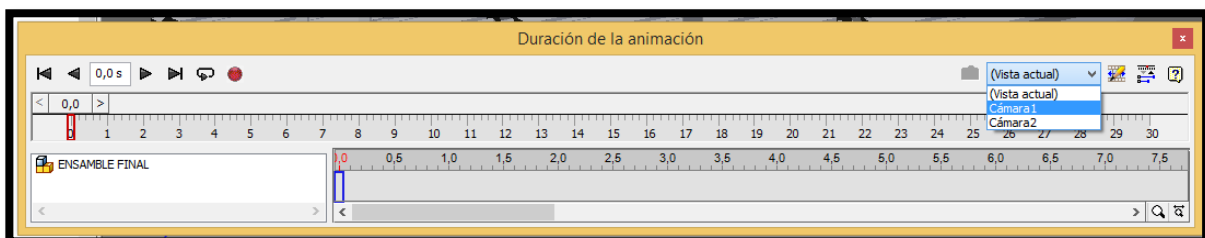


Figura 5.31 Selección de cámara

El procedimiento para seleccionar el tiempo donde se desea utilizar la cámara, así como agregar rotación y opciones de vista de la cámara, se efectúa seleccionando la opción cámara en el navegador de operaciones (Figura 5.32).

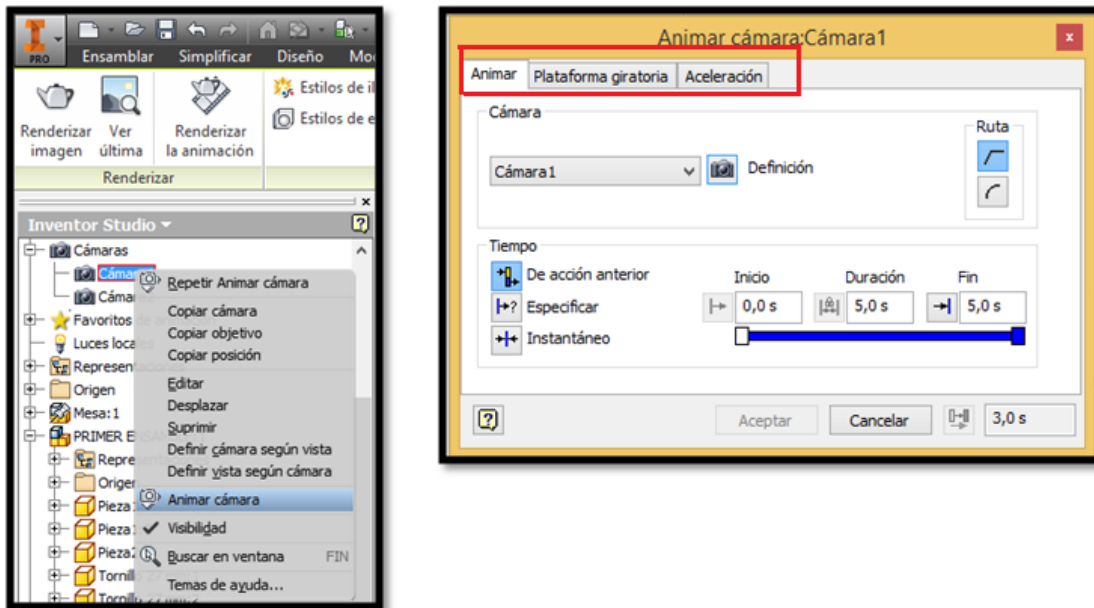


Figura 5.32 Configuración de cámara

Animada la simulación y definidas las cámaras, se creará la producción del video. Se elige en la barra de herramientas la opción Productor de vídeo. En la ventana emergente se colocan las cámaras arrastrándolas desde el menú tomas, se eligen las transiciones entre cámaras, se editan los tiempos y finalmente se obtiene el video (Figura 5.33).

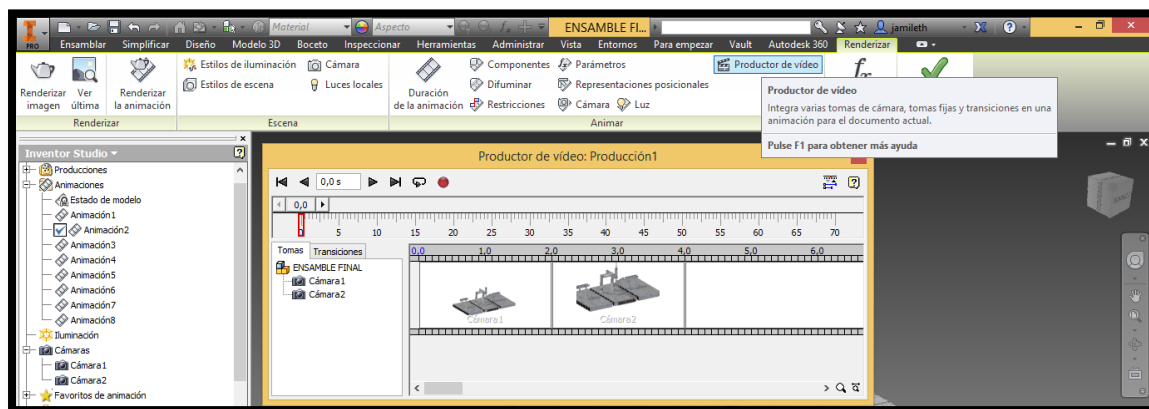


Figura 5.33 Producir vídeo

El programa tiene la opción de Renderizar la animación, acción que consigue videos con un acabado más real. Desde la barra de herramientas principal se selecciona la opción



Renderizar Animación. En la nueva ventana se elige el formato de salida del video, la duración y la velocidad de transición de imágenes (Figura 5.34).

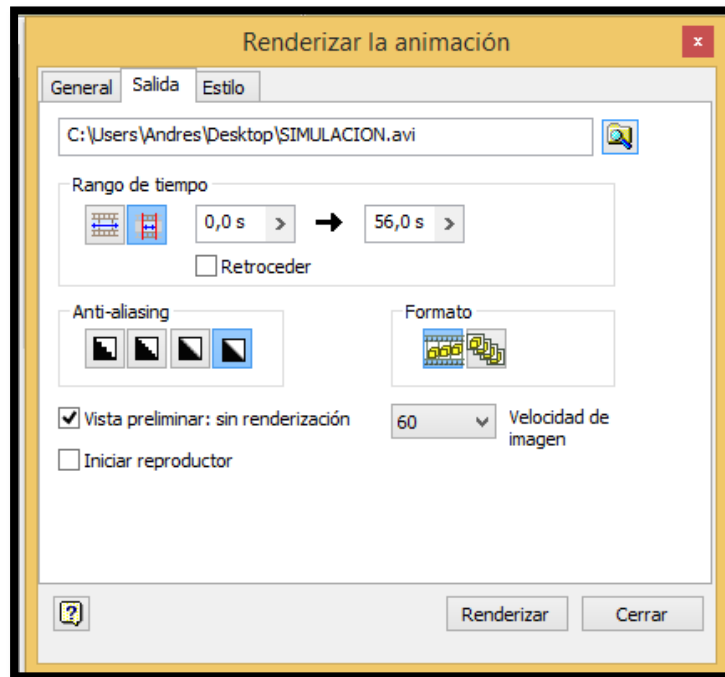


Figura 5.34 Renderizar animación

En la Figura 5.35 se observa la renderización en proceso de la Estación de acarreo de piezas.

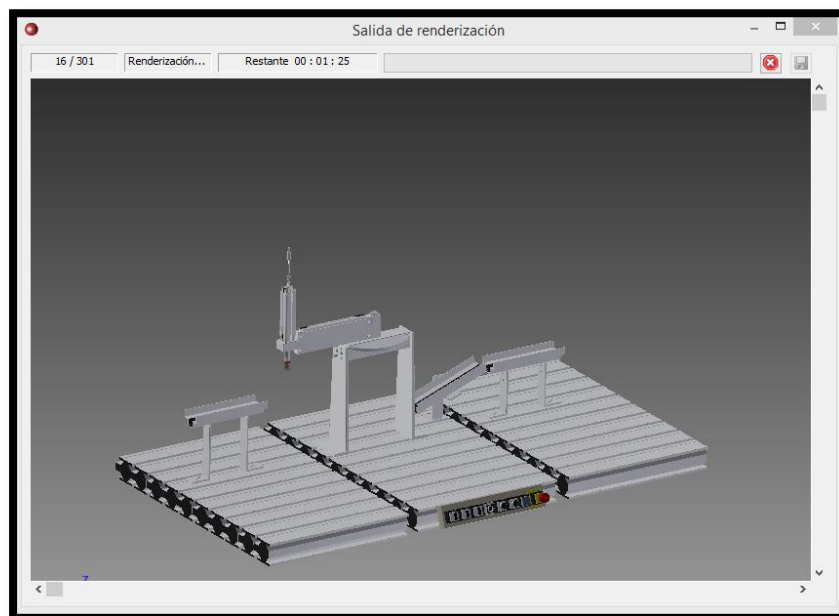


Figura 5.35 Video renderizado Estación de acarreo de piezas

### 5.3.5 Entorno de presentación

Es un módulo para la creación de presentaciones a partir de ensambles hechos previamente, permite realizar vistas explosionadas de las piezas y animaciones de las mismas.

Para la presentación del despiece de la Estación de acarreo de piezas, se utilizó un software especializado en presentaciones, que hace parte del paquete de Inventor Autodesk.

Autodesk Inventor Publisher 2015 es un programa útil para realizar presentaciones de un proyecto. Mediante este programa se obtiene documentación técnica en 2D, en él se puede efectuar documentos informativos como por ejemplo: un manual de una máquina, ya que este programa tiene la opción insertar textos, imágenes, flechas, resaltar elementos, etc., que te permite completar así las instrucciones de una máquina, pieza o ensamblaje específico.

También se puede crear documentación de un producto en 3D. El programa crea presentaciones denominadas Instantáneas, que el usuario va realizando a sus necesidades y el conjunto de ellas expuestas en un tiempo determinado crean una historia completa de un producto. Como resultado se obtiene una animación interactiva, con anotaciones, dimensiones, detalles, cuadros de lista de materiales, entre otras.

El software te permita la importación de datos multiCAD, es decir importar datos 3D de Inventor Fusión, DWG, Autodesk 123D, SolidWorks, Google SketchUp, CATIA, UG NX, ProENGINEER, DWG®, DWF, STEP, IGES, FBX, OBJ, y software SAT, así como bloques en 2D a partir de archivos de software de AutoCAD.

Por último para concluir la presentación 2D o 3D, el programa te da la opción de publicar el trabajo hecho en distintos formatos ya sea en formato móvil o directamente a video, la tecnología Adobe Flash, MP4, WMV, 3D PDF, SVG, Word y el software de Microsoft PowerPoint.

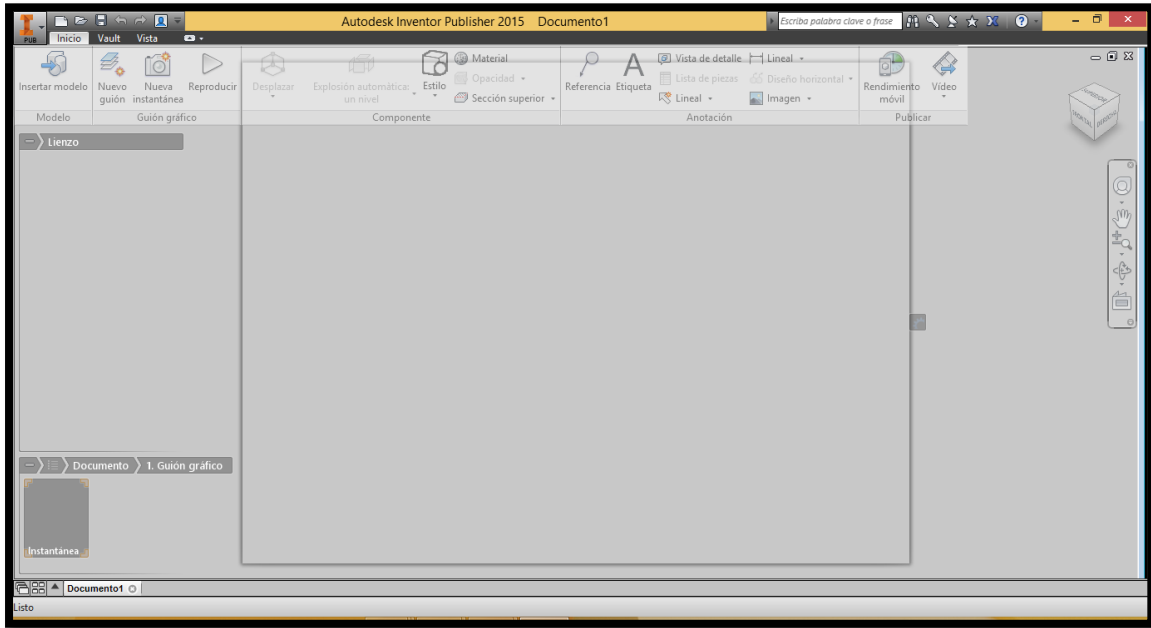


Figura 5.36 Interfaz de Autodesk inventor Publisher

Se detalla a continuación cómo se realizó la presentación de la Estación de acarreo de piezas:

Una vez en la interfaz de usuario, se selecciona la opción “nuevo” (Figura 5.37).

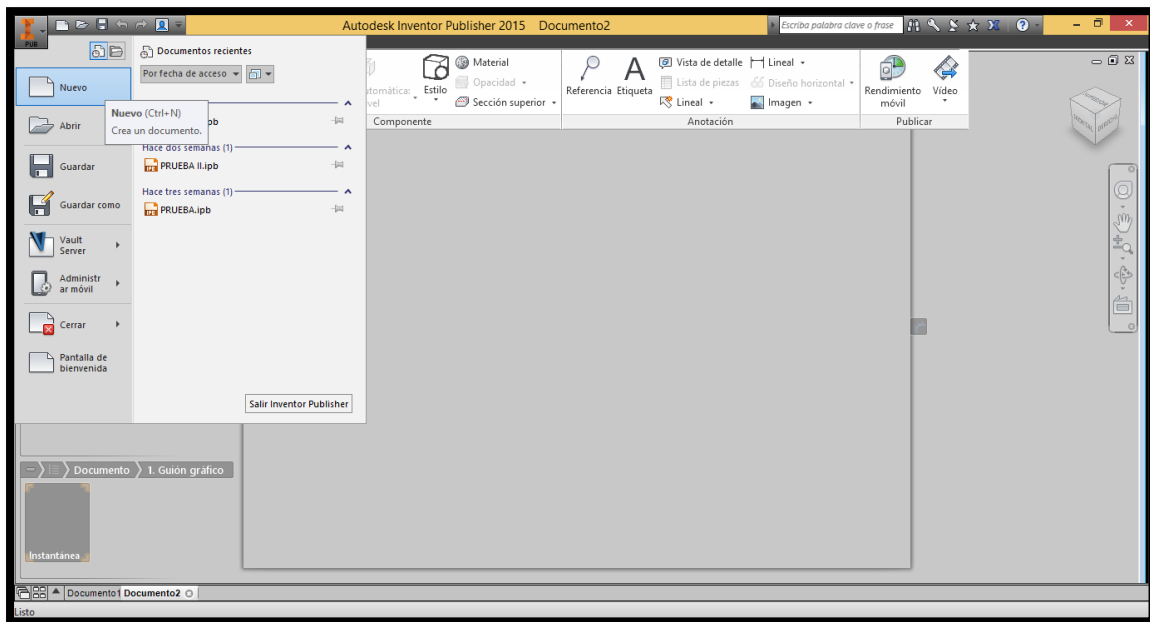


Figura 5. 37 Nuevo proyecto

En la ventana emergente, se escoge el documento a utilizar. Inventor Publisher trabaja con los archivos .ipt y los .iam de Inventor, además de archivos de otros programas como AutoCad, Catia, Solidworks, etc.

Para el caso de Estación de acarreo de piezas, se escoge el ensamble final publisher formato de archivo .iam (Figura 5.38).

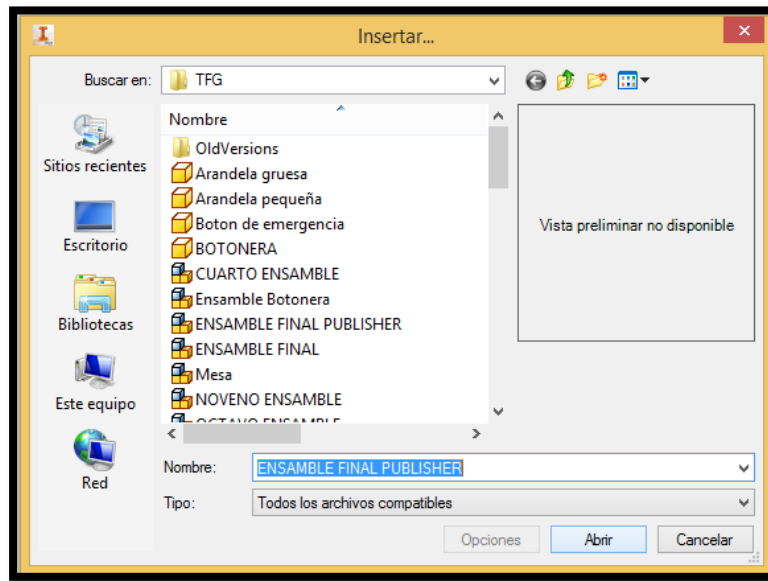


Figura 5.38 Elección de archivo

La presentación es un conjunto de tomas instantáneas (Figura 5.39), que reproducidas en un determinado tiempo, se logra recrear la animación del despiece de elementos. El usuario debe indicar en cada diapositiva que maniobra realiza, por ejemplo, realizar un despiece, cambiar el color de las piezas, quitar la visibilidad de un elemento, cambiar el ángulo de la cámara, agregar texto, etc.

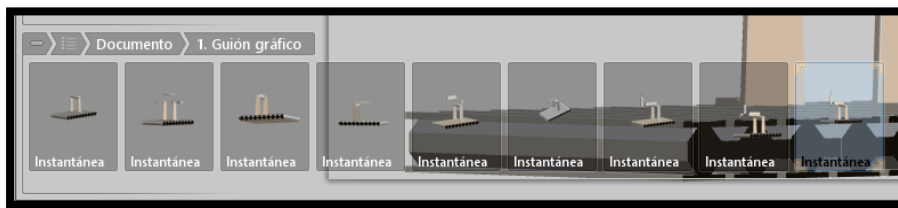


Figura 5.39 Grupo de Instantáneas

Para agregar una diapositiva se hace mediante la opción nueva instantánea, de la barra de herramientas (Figura 5.40).



Figura 5.40 Opción nueva instantánea

Para realizar el despiece: visualizando el ensamble seleccionado y pulsando clic derecho del ratón sobre el entorno de trabajo, aparecerá en el menú la opción explosión manual (Figura 5.41).

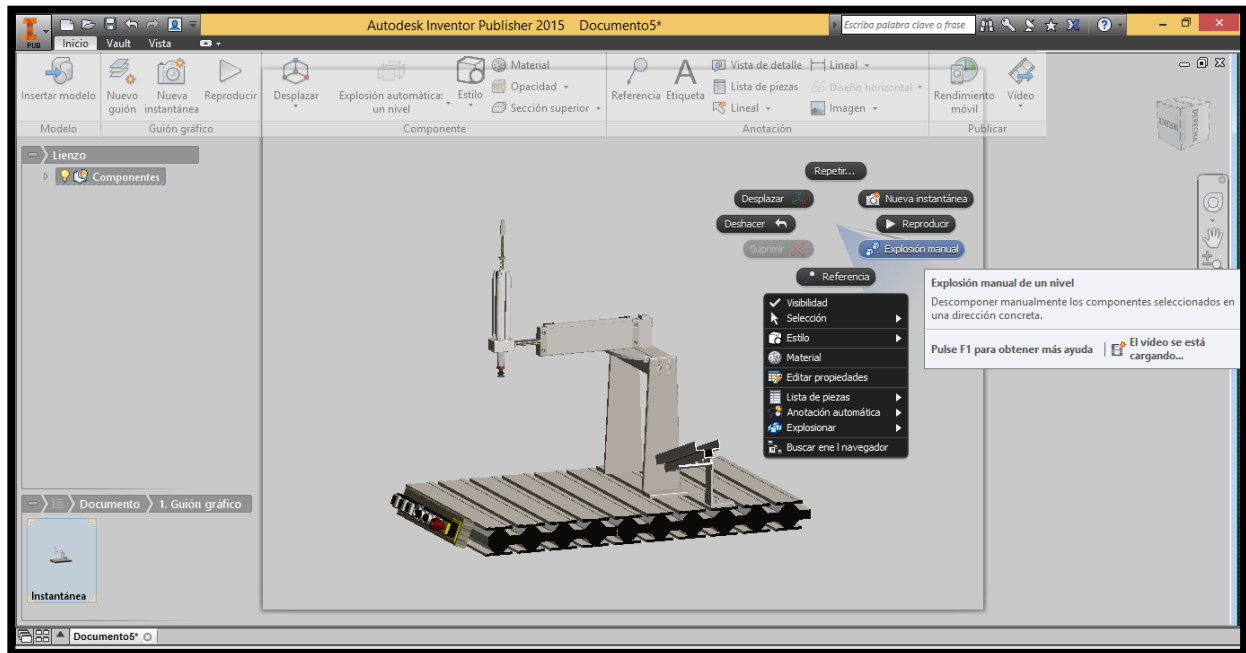


Figura 5.41 Opción explosión

La opción permite al usuario descomponer el ensamblaje de la manera que mejor le perezca, se elige la dirección del despiece, seleccionando y arrastrando cada elemento (Figura 5.42).

Esta opción se aplica en cualquiera de las diapositivas cuando sea conveniente.

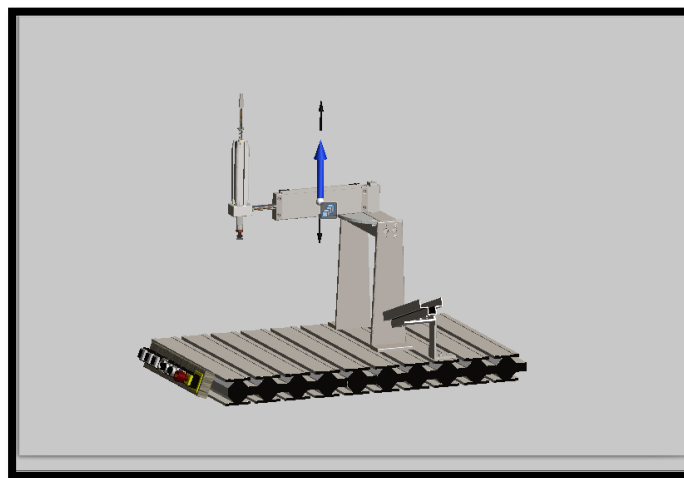


Figura 5.42 Despiece de elemento

El programa tiene la opción de restaurar el ensamblaje de manera rápida: seleccionando todo el ensamblaje, clic derecho en el entorno de trabajo/restablecer inicio (Figura 5.43).

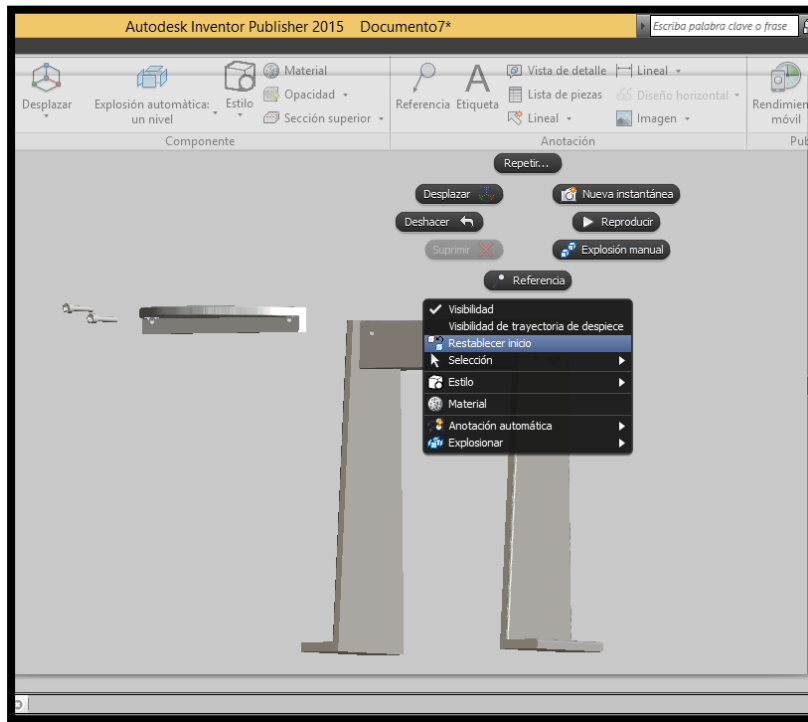


Figura 5.43 Restaurar ensamblaje

Para reproducir la representación en cualquier instante se realiza desde la barra de herramientas (Figura 5.44).

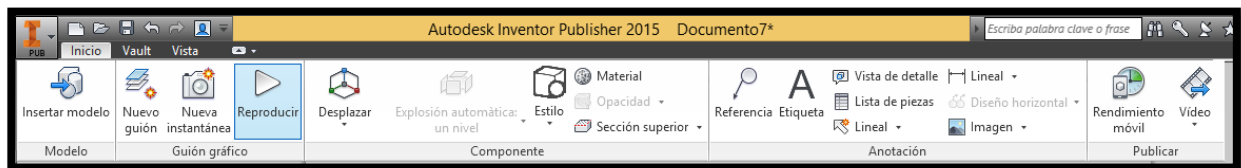


Figura 5.44 Reproducir presentación

Para modificar el tiempo de transición entre cada diapositiva se pulsa el botón derecho del ratón sobre el menú documento/ mostrar sincronización (Figura 5.45).

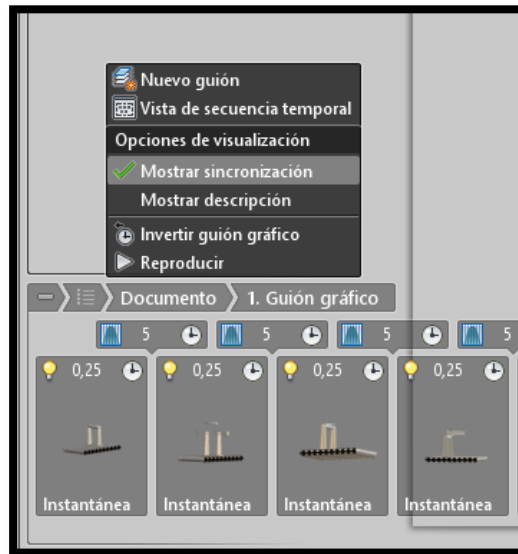


Figura 5.45 Modificar tiempo de transición

Si el usuario desea activar alguna cámara en una instantánea específica: situarse en la instantánea, clic derecho escoger opción activar (Figura 5.46).

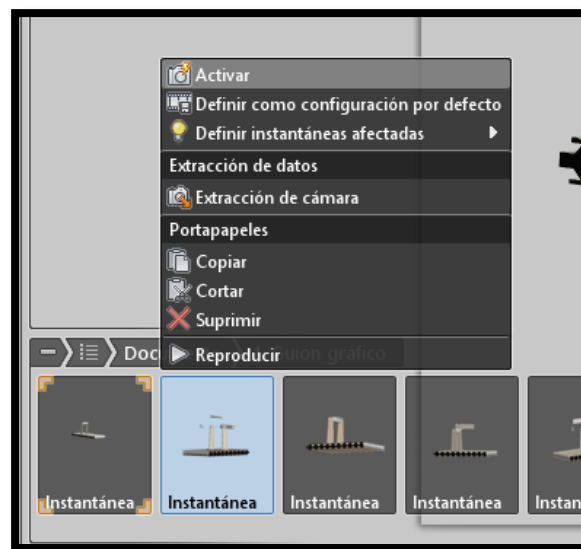


Figura 5.46 Activar Cámara

Una vez terminada la presentación, en la barra de herramientas se localiza la opción video, en esta opción se encuentran los diferentes formatos en los que se puede guardar el trabajo realizado (Figura 5.47).

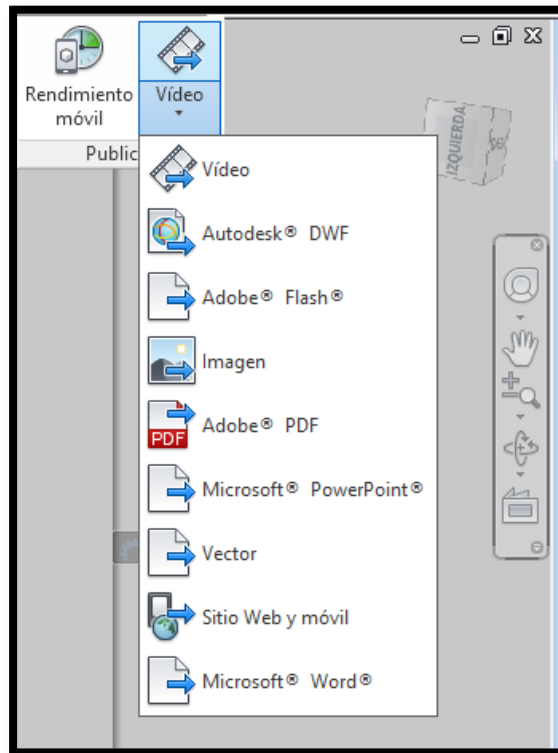


Figura 5.47 Formatos de la presentación

### 5.3.6 Entorno de documentación

Es un módulo para la delineación de planos a partir de piezas, conjuntos soldados o mediante herramientas de dibujo. Cuando se realiza un plano a partir de una pieza o un conjunto se crea un vínculo entre el archivo que contiene la pieza y el archivo que contiene el plano, de manera que si posteriormente se modifica la pieza, el plano automáticamente también se modifica.

El software Inventor genera de forma semiautomática las especificaciones técnicas de cualquier pieza o ensamblaje creado. Los ficheros del entorno de documentación se almacenan con la extensión .idw (*Inventor Drawing*).

Procedimiento para la obtención de especificaciones técnicas de la Estación de acarreo de piezas:



En la interfaz del programa se escoge la opción dibujo (Figura 5.48)

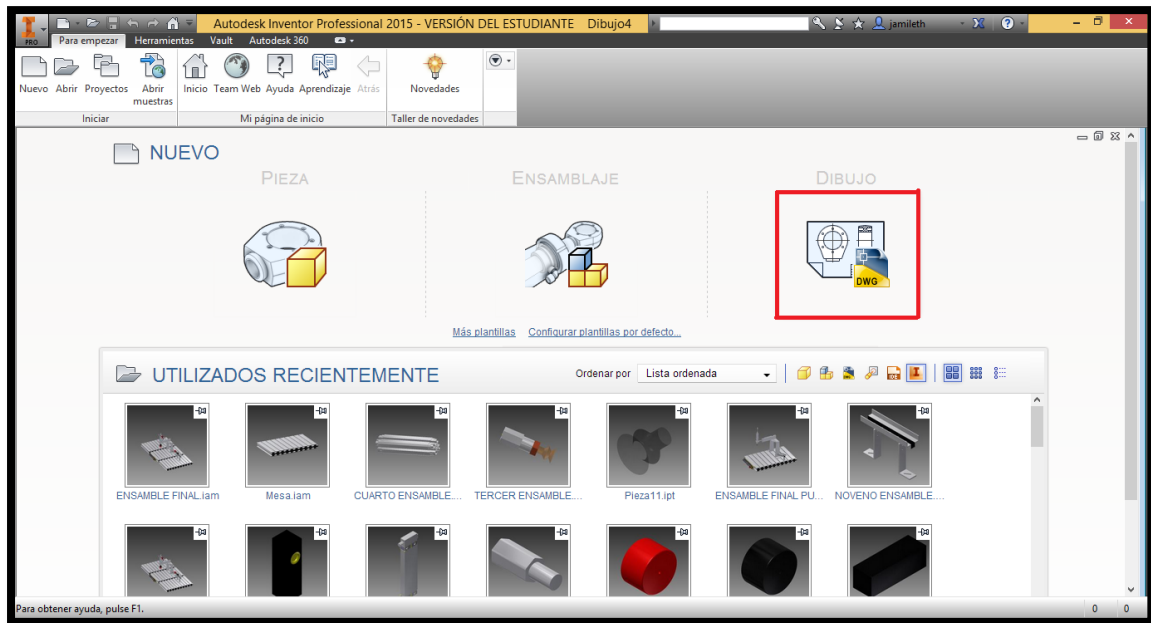


Figura 5.48 Opción dibujo

En el interfaz de trabajo, se aprecia la plantilla que sale por defecto, formato A3, marco incluido y con un cajetín normalizado ISO. (Figura 5.49)

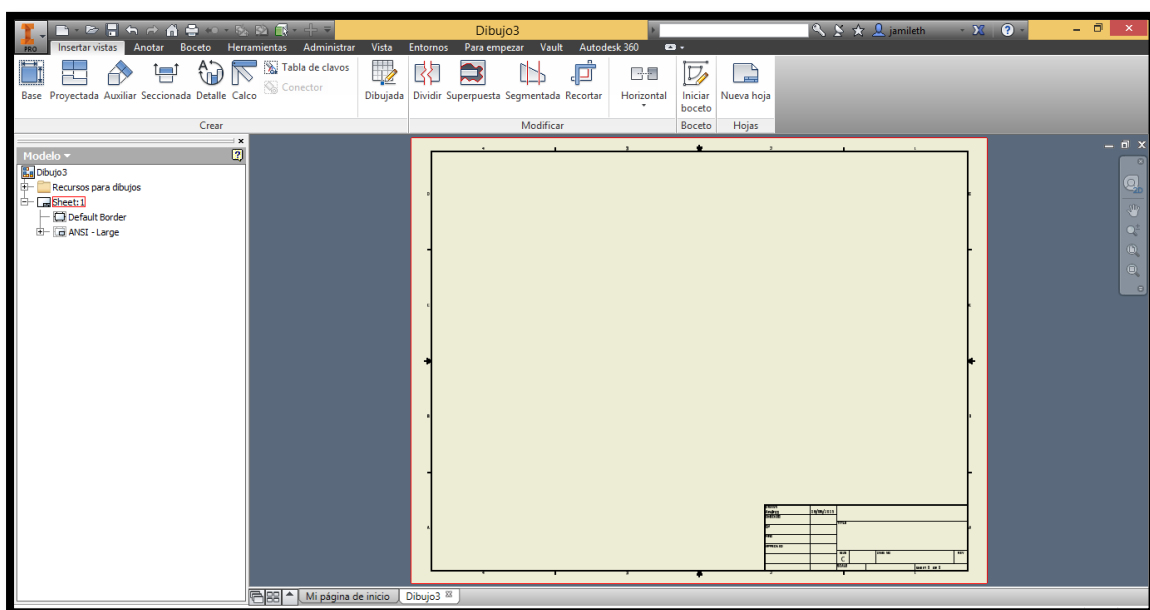


Figura 5.49 Plantilla predeterminada

En el navegador de operaciones se encuentra una capeta que se llama Recursos para dibujos, seleccionándola, aparece datos como el formato de hoja, marcos y cajetines. Estos elementos pueden ser modificados a gusto del usuario. Para ello se ubica el elemento y con clic derecho del ratón aparece a opción editar (Figura 5.50)

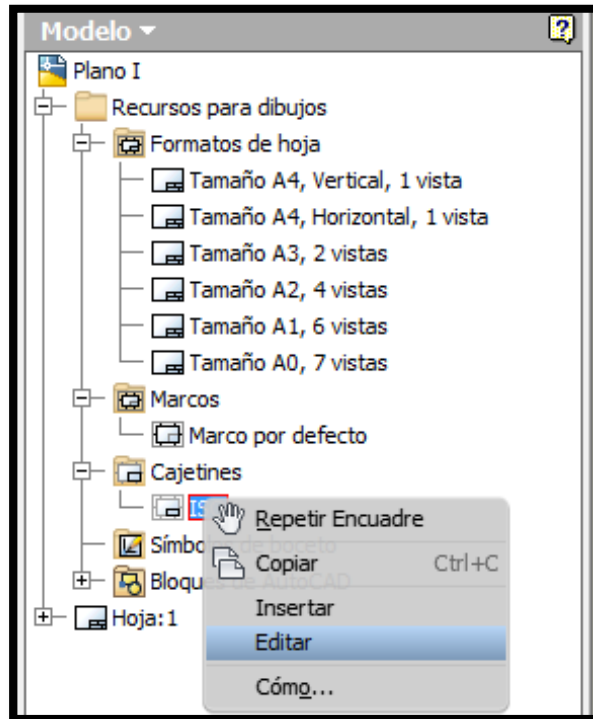


Figura 5.50 Editar cajetín

Cuando se elige esta opción en la interfaz de trabajo, se abre el cajetín por defecto, el cual puede modificarse con las herramientas típicas de un boceto (Figura 5.51).

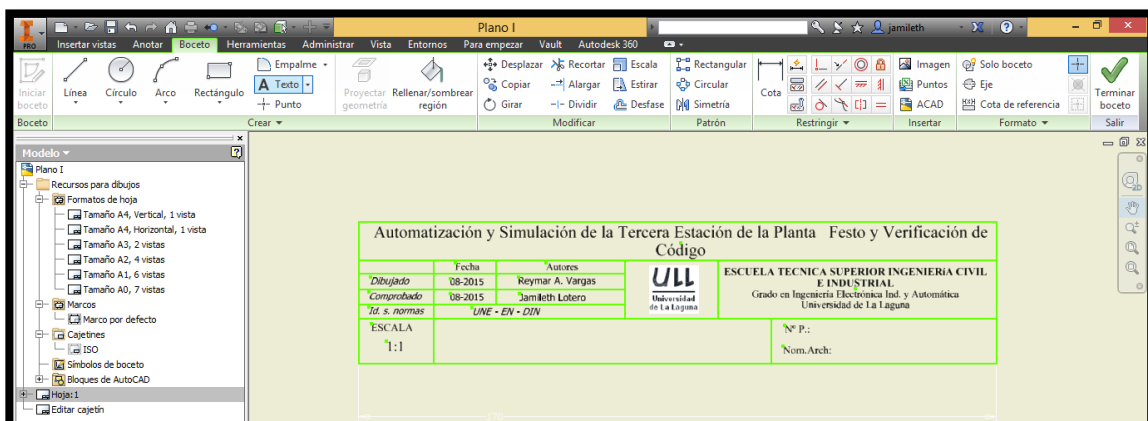


Figura 5.51 Realizar nuevo cajetín

Para obtener el plano de una pieza, se debe seleccionar del menú de herramientas la opción de base (Figura 5.52)

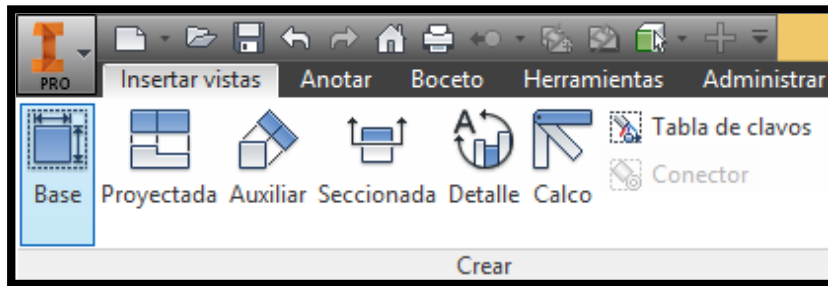


Figura 5.52 Obtener Plano de una pieza

Se abrirá un cuadro de dialogo donde, se escoge la pieza, el usuario debe indicar la escala que desea, el tipo de vista y el estilo de la pieza (Figura 5.53).

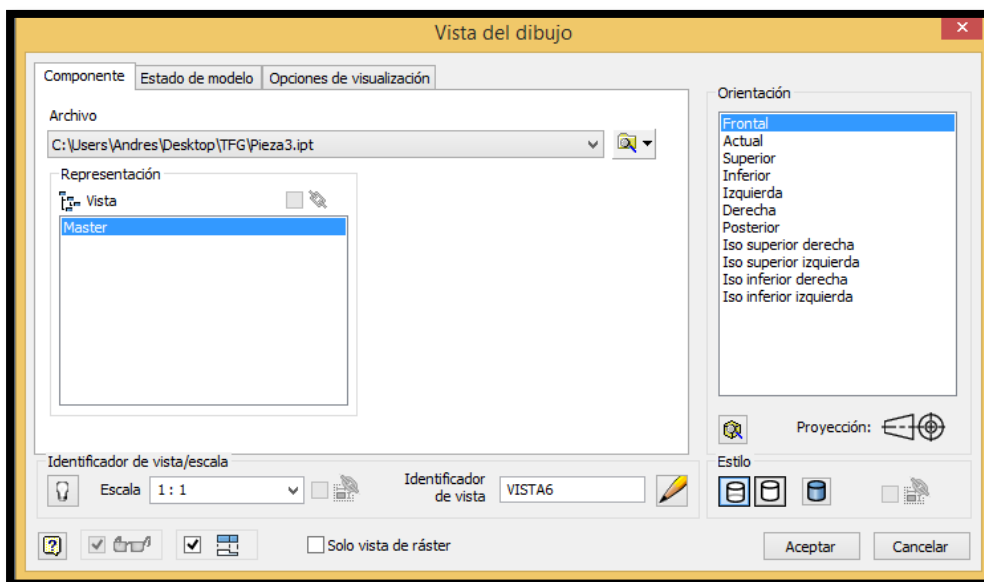


Figura 5.53 Configurar plano

El programa crea automáticamente la vista de base seleccionada, en el mismo documento se puede repetir acción anterior y obtener un conjunto de las vistas más relevantes de la pieza (Figura 5.54).

Creado el plano con las vistas deseadas, el usuario tiene la opción de exportarlo a PDF.

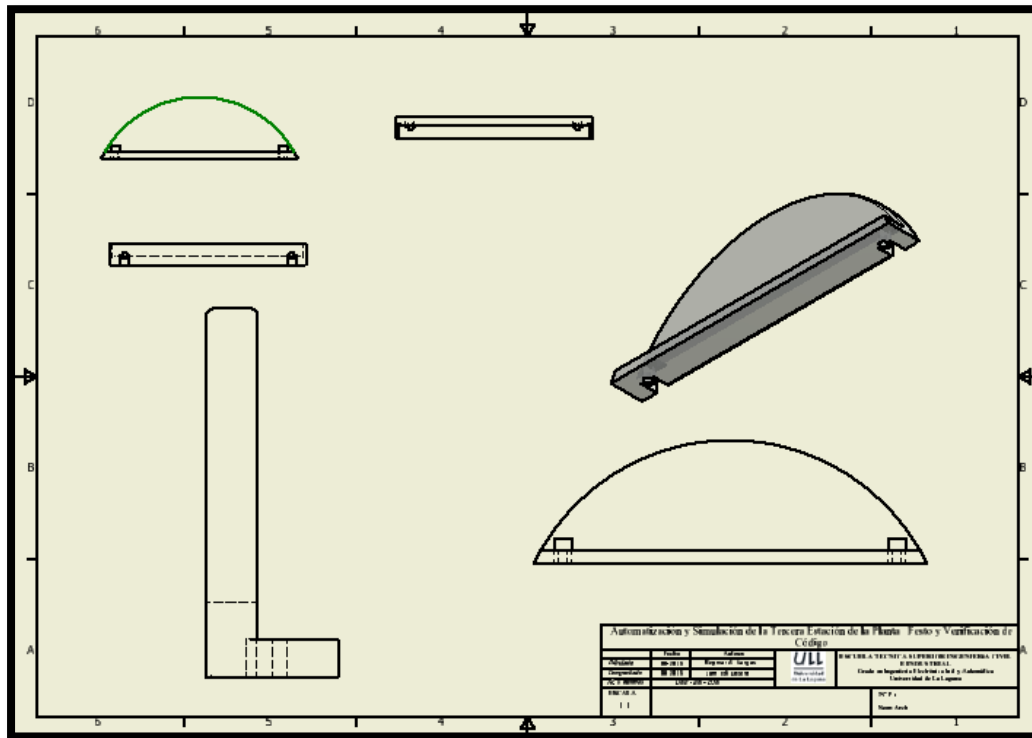


Figura 5.54 Creación de vistas

Este módulo cuenta con herramientas más avanzadas para documentos donde sea necesario una ficha técnica más informativa. Para acceder a estas herramientas se realiza desde la interfaz de trabajo, seleccionando el menú anotar, se despliegan todas estas herramientas (Figura 5.55)

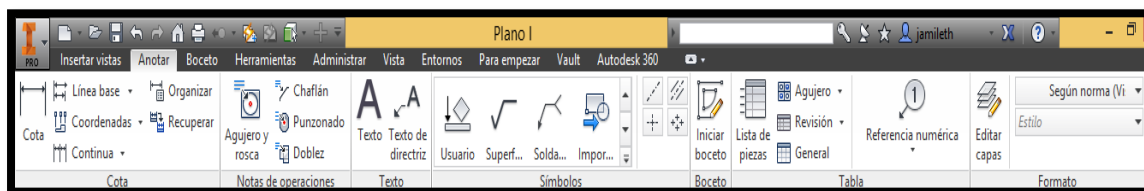


Figura 5.55 Opciones para los planos

## **Capítulo 6. Estudio del Software educativo para el aprendizaje en automatización.**

### **6.1 Introducción**

Hoy en día la docencia busca cómo mejorar el aprendizaje de los alumnos, así como la motivación de los mismos por aprender. Por este motivo cada vez más se utilizan software enfocados a la enseñanza, con el fin de mejorar la calidad de la misma.

Así pues, se puede decir que un software educativo es una herramienta que se emplea para el aprendizaje, con el fin de ayudar a los alumnos (desde niños hasta universitarios) para que adquieran conceptos, principios y habilidades que resultan complejos de entender o bien para mejorar los contenidos dados en alguna asignatura, apoyándose de distintos medios como textos, imágenes, vídeos, sonidos...

Los estudiantes utilizan programas especializados para la materia a tratar, donde disponen de la información necesaria y actividades. Esto ofrece ventajas como mejorar la calidad del aprendizaje, los alumnos aprenden de manera autónoma, acceso a los materiales vía internet, etc.

#### **6.1.1 Objetivo**

El software educativo se realiza para la asignatura Automatización Industrial Avanzada del cuarto curso del Grado en Ingeniería Electrónica Industrial y Automática.

Desafortunadamente en un cuatrimestre no es posible adquirir todos los conocimientos posibles sobre una materia tan amplia como lo es la automatización, es por ello que se suelen dejar pendientes temas que podrían ser interesantes de tratar para el alumno.

Un ejemplo de esto es la automatización en el entorno de CoDeSyS, que es la herramienta de desarrollo basada en IEC 61131-3 más extendida en Europa, utilizada por más de 250 fabricantes de dispositivos de diferentes sectores industriales.

Como la universidad de la Laguna cuenta para la planta Festo con autómatas de Siemens, sólo se imparten conocimientos sobre la programación en entorno de Step 7 Microwin y Step 7 lite.

Con la elaboración de este programa se pretende mejorar los conceptos adquiridos en dicha asignatura como la programación en Step 7 Microwin, familiarizar a los estudiantes con el estándar IEC 61131-3, motivar a los alumnos para que aprendan de forma autónoma la programación de CoDeSys para el lenguaje de diagrama de contactos y para el de listas de instrucciones. Esto permitirá a los estudiantes hacer una comparación de este programa con Microwin.

Para que el aprendizaje sobre estos temas sea más dinámico, se les facilitará a los estudiantes el material necesario para reforzar los conceptos aprendidos en clase, y para evaluar sus conocimientos en la materia se realizará un cuestionario a modo de test con conceptos teóricos, ejercicios y comparaciones.

## **6.2 Estudio y elección del programa**

### **6.2.1 Programación en Python**

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web.

Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables. Es fácil de usar, ofreciendo mayor estructura y soporte para programas grandes que lo que lo que pueden ofrecer los scripts de Unix o archivos por lotes.

Por otro lado, Python ofrece mucho más chequeo de error que C, y siendo un lenguaje de muy alto nivel, tiene tipos de datos de alto nivel incorporados como arreglos de tamaño flexible y diccionarios. Permite separar un programa en módulos que pueden reusarse en otros programas de Python.

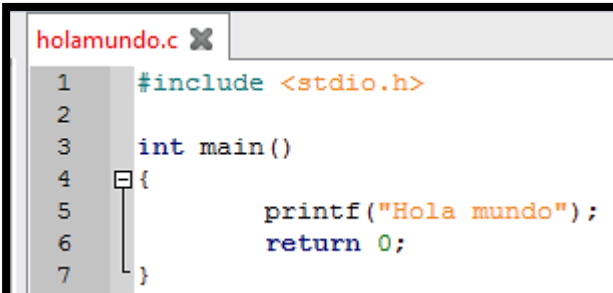
Python es un lenguaje interpretado, lo cual puede ahorrar mucho tiempo durante el desarrollo ya que no es necesario compilar ni enlazar. El intérprete puede usarse interactivamente, lo que facilita experimentar con características del lenguaje, escribir programas descartables, o probar funciones cuando se hace desarrollo de programas de abajo hacia arriba.

Los programas en Python son típicamente más cortos que sus programas equivalentes en C, C++ o Java por varios motivos:

- Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola instrucción.
- La agrupación de instrucciones se hace por sangría en vez de llaves de apertura y cierre.
- No es necesario declarar variables ni argumentos. [13]

En el siguiente ejemplo se puede observar la sencillez de este lenguaje si es comparado con C o C++. El primer programa que realiza un estudiante dedicado a la programación es el conocido “hola Mundo”.

Hola Mundo en C:



```
holamundo.c x
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Hola mundo");
6      return 0;
7  }
```

Figura 6.1 Programación en C

Hola Mundo en C++:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6
7      cout << "Hola Mundo" << endl;
8
9      return 0;
10
11 }
```

Figura 6.2 Programación en C++

Hola Mundo en Python:

```
1  print "Hola mundo";
```

Figura 6.3 Programación en Python

Mientras que para el lenguaje en C se utilizaron 7 líneas de código y en C++ 11 líneas de código, en Python sólo fue necesario utilizar una línea de código ya que no se tienen que declarar las librerías como en los otros dos casos, ni se utiliza la función *main ()* que define la función principal del programa. Sólo con mirar este ejemplo se llega a la conclusión de que manejar Python es más sencillo que otros lenguajes.

### 6.2.2 Python para la elaboración del programa

En un principio se planteó la posibilidad de elaborar el programa educativo utilizando un lenguaje de programación que fuese sencillo como lo es Python.

El programa en cuestión se planteó de la siguiente manera:

1. Elaboración de un cuestionario a modo de test donde se plantean distintas situaciones por las que podría pasar la estación de acarreo de piezas, teoría de programas como CoDeSyS y Microwin, programación en KOP y AWL...
2. EL cuestionario constaría aproximadamente de 50 preguntas.
3. Dependiendo de la respuesta del alumno, se inserta un vídeo del modelo en 3D creado por Inventor.



4. Cada respuesta incorrecta debe incluir una explicación para que los alumnos comprendan mejor el concepto del que se les habla.

Antes que nada, para poder manejar un lenguaje de programación es necesario tener un mínimo de conocimiento sobre él. En este caso se tuvo que partir de cero para poder llegar a una conclusión: Python no es tan sencillo como parece.

Es verdad que sólo son un par de preguntas, sin embargo su complicación no está en la elaboración de éstas sino en el momento de insertar un archivo multimedia en Python (si no se tiene una buena base).

A continuación se presenta el estudio realizado en el proyecto y el motivo por el cual no se pudo utilizar el lenguaje Python para el cuestionario.

### ¿Cómo elaborar una pregunta con las características planteadas anteriormente?

Para programar en Python es necesario tener un editor con el que trabajar e instalar y configurar Python correctamente para poder trabajar con él. El editor de texto que se emplea es Geany.

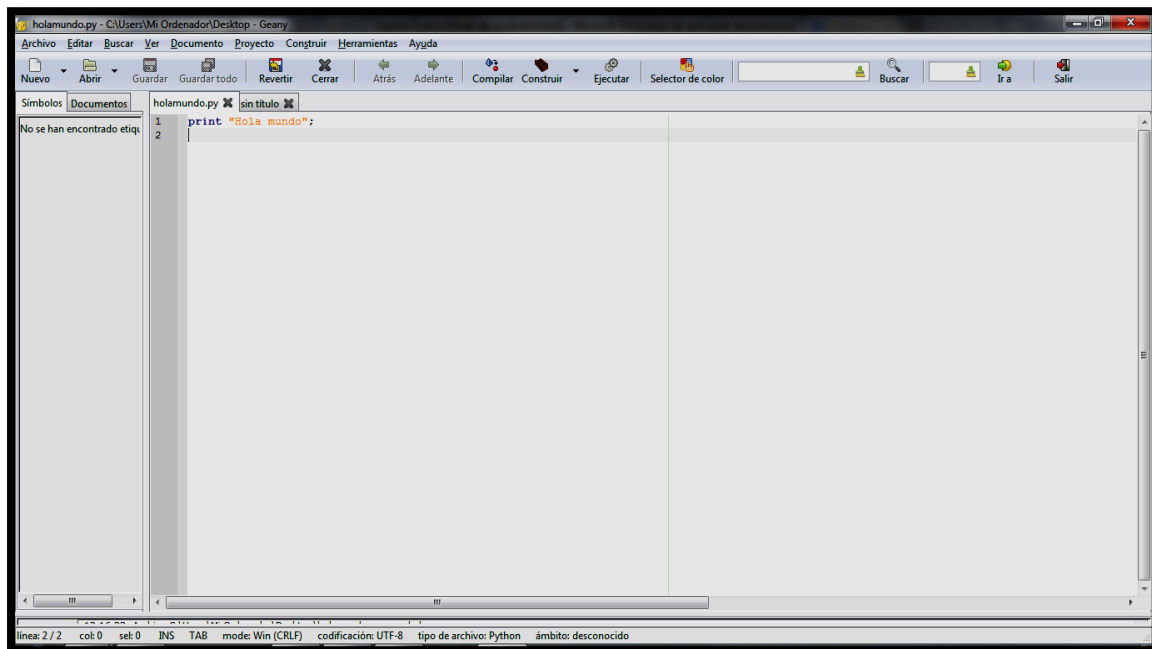


Figura 6.4 Editor de texto Geany

A continuación se adjunta una de las preguntas del cuestionario realizada en Geany:

```
Pregunta.py x
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  print( ' Definne el concepto de marca : ')
5  print('Son variables auxiliares que pueden retener un determinado resultado -> 1')
6  print('Se reconocen por la letra M -> 2')
7  print('Se asignan de la misma manera que las salidas -> 3')
8  print('Todas son correctas -> 4\n')
9
10 n=int(input('seleccione la respuesta que considere correcta: \n'))
11 if n!=4 :
12     print('respuesta incorrecta, todas las opciones son correctas \n')
13 else:
14     print('correcto \n')
15
```

Figura 6.5 Pregunta elaborada en Python

Ésta es una pregunta sencilla de teoría en la que no se insertan archivos multimedia. Se plantea la pregunta y varias opciones. El usuario sólo debe introducir el número de la respuesta que considere correcta. Una vez elaborada la pregunta se procede a ejecutarla desde la consola de Windows (CMD):

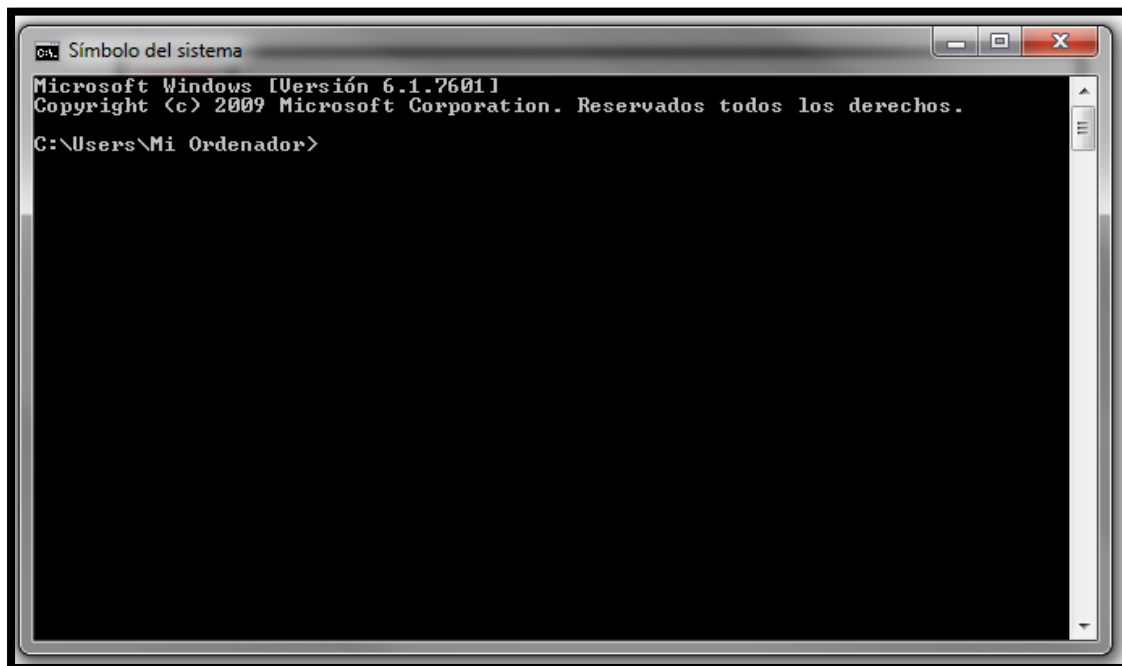


Figura 6.6 Consola de Windows

Se llama desde la consola la pregunta que se elaboró en Geany:

```

C:\Users\Mi Ordenador>cd desktop
C:\Users\Mi Ordenador\Desktop>python pregunta.py
    
```

Figura 6.7 Ejecución del programa

La pregunta queda de la siguiente manera:

```

C:\Users\Mi Ordenador\Desktop>python pregunta.py
Definne el concepto de marca :
Son variables auxiliares que pueden retener un determinado resultado -> 1
Se reconocen por la letra M -> 2
Se asignan de la misma manera que las salidas -> 3
Todas son correctas -> 4

seleccione la respuesta que considere correcta:
    
```

Figura 6.8 Pregunta en Python

El alumno debe elegir una respuesta del 1 al 4. Si el alumno no selecciona la respuesta correcta, en este caso la cuarta, aparecerá la siguiente línea:

```

C:\> Símbolo del sistema
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Mi Ordenador>cd desktop
C:\Users\Mi Ordenador\Desktop>python pregunta.py
Definne el concepto de marca :
Son variables auxiliares que pueden retener un determinado resultado -> 1
Se reconocen por la letra M -> 2
Se asignan de la misma manera que las salidas -> 3
Todas son correctas -> 4
seleccione la respuesta que considere correcta: 2
Respuesta incorrecta, todas las opciones son correctas
    
```

Figura 6. 9 Respuesta incorrecta

Si la respuesta es correcta:

```

C:\> Símbolo del sistema
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Mi Ordenador>cd Desktop
C:\Users\Mi Ordenador\Desktop>python pregunta.py
Definne el concepto de marca :
Son variables auxiliares que pueden retener un determinado resultado -> 1
Se reconocen por la letra M -> 2
Se asignan de la misma manera que las salidas -> 3
Todas son correctas -> 4
seleccione la respuesta que considere correcta: 4
Su respuesta es correcta
    
```

Figura 6.10 Respuesta correcta

### 6.2.3 Inconvenientes

Como se puede apreciar, elaborar un cuestionario de la forma que se hizo anteriormente no es complicado. Sin embargo, el programa que se requiere para el proyecto no se corresponde exactamente con la demostración anterior.

Para que los alumnos puedan comprender mejor la pregunta que se les plantea, en algunas cuestiones resulta necesario apoyarnos de archivos multimedia, como partes del código de la estación 3, entorno de CoDeSyS, ejemplos del lenguaje KOP y AWL, vídeos de la simulación en 3D de la estación, etc.

Una pregunta en la que se necesite insertar estos archivos puede diseñarse de la siguiente manera:

1. Se plantea la pregunta al alumno sobre un estado concreto del brazo de la estación.
2. Se les dan las posibles opciones.
3. Si el alumno escoge la respuesta correcta, saldrá un mensaje diciendo que ha elegido la respuesta correcta y un video que muestre el brazo realizando esta acción.

Si el alumno responde mal, saldrá un mensaje diciendo que la respuesta es incorrecta, y un video demostrando cómo se ve el brazo realizando el movimiento que éste eligió como correcto.

Si en Python hubiese una opción que permitiese insertar imágenes o videos, el cuestionario sería sencillo de elaborar.

Se intentó insertar enlaces de videos para que aparecieran en las respuestas, con sólo copiar y pegar un enlace, pero no funcionó.

Por ello se estudiaron algunos módulos de Python para una interfaz de usuario (programas que permiten interactuar con el usuario por medio de botones, menús...). Uno de estos módulos es **Tkinter**. Gracias a éste se pueden crear ventanas, botones, menús, agregar etiquetas, etc.

```

tk.py ✖
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4
5  from Tkinter import *
6
7  def saludar():
8      print 'Hola'
9
10 w = Tk()
11
12 l = Label(w, text='Hola programa')
13 l.pack()
14
15 b1 = Button(w, text='Saludar', command=saludar)
16 b1.pack()
17
18 b2 = Button(w, text='Salir', command=exit)
19 b2.pack()
20
21 w.mainloop()
22

```

Figura 6.11 Ejemplo de Tkinter

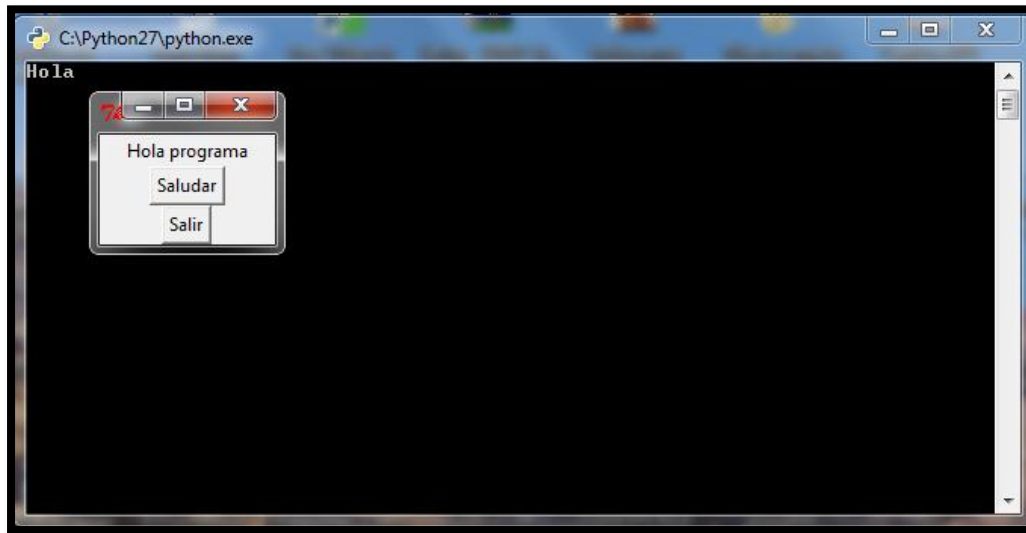


Figura 6.12 Ventana creada desde Tkinter

En este ejemplo el usuario dispone de dos botones “saludar” y “salir”, si presiona la opción “saludar” en la consola aparece el mensaje “hola” y si pulsa “salir” se cierra el programa.

Este programa se parece un poco más a lo que se quiere para el proyecto, sin embargo sigue el problema de insertar un archivo multimedia.

Luego se estudió una biblioteca de Python que se conoce como Pygame. Esta librería encapsula a SDL (*Simple Directmedia Layer*). Se pueden manipular gráficos, audio, sonido, y está orientada al desarrollo de video juegos. [14]

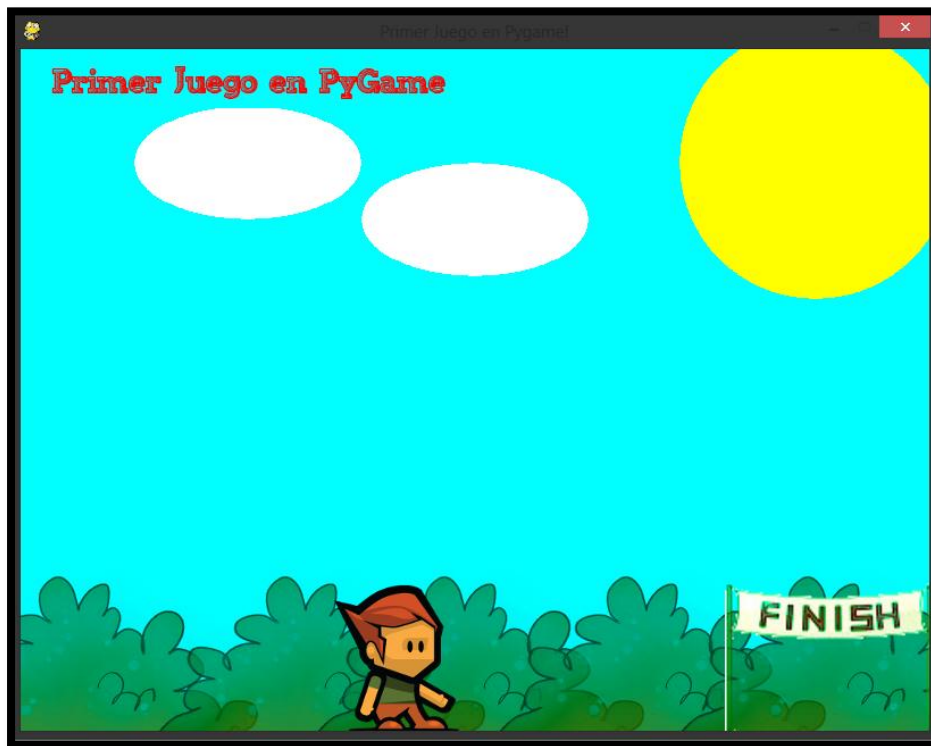


Figura 6.13 Primer juego de Pygame

En este ejemplo del video juego, el muñeco se mueve hacia la derecha e izquierda hasta llegar a la meta. Cuando él llega se termina el juego. Para crearlo hay que definir el tamaño de la ventana, buscar una imagen del muñeco, el arbusto. El sol y las nubes se definen en Pygame, al igual que el color del cielo. Parece sencillo de elaborar pero la sintaxis que se utiliza no es fácil de entender para aquellas personas que no se dediquen a la programación.

A continuación se muestra el código del juego tomado como ejemplo, extraído del tutorial que se encuentra en el enlace [15]:

```
1
2 import pygame, sys
3 from pygame.locals import *
4
```

importamos los modulos necesarios para nuestro juego

```
1
2 pygame.init()
3
```

Inicializamos el modulo de pygame, necesario antes de empezar a utilizar cualquier cosa de PyGame

```
1
2 FPS = 30
3 frecuencia = pygame.time.Clock()
4
```

Creamos una variable con los fotogramas por segundo, y otra con la que modificaremos la frecuencia de reloj de nuestro juego

```
1
2 PANTALLA = pygame.display.set_mode((800, 600),0,32)
3 pygame.display.set_caption('Primer Juego en Pygame!')
4
```

Creamos nuestro objeto para manejar la pantalla, establecemos la resolucio de pantalla y profundidad, establecemos tambien el titulo de ventana

```
1
2 CIELO = (0,255,255)
3 NUBE = (255,255,255)
4 SOL = (255,255,0)
5 ROJO = (255,0,0)
6
```

Establecemos variables para los colores que utilizaremos, utilizamos el modelo RGB para crear los colores

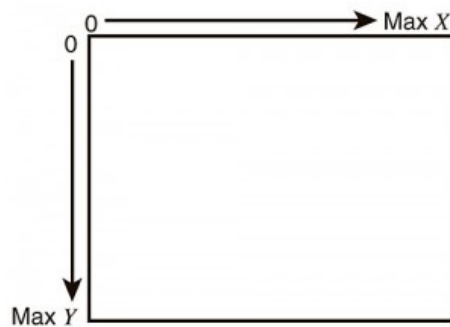
```
1
2 PERSONAJE = pygame.image.load("personaje.png")
3 ARBUSTO = pygame.image.load("arbusto.png")
4 META = pygame.image.load("meta.png")
5
```

Creamos variables que almacenaran las imagenes que queremos mostrar en nuestro juego, estas imagenes estan en el archivo comprimido que esta en requerimientos, estas son la imagen de nuestro personaje principal, una imagen de un arbusto que repetiremos algunas veces, la imagen de la meta.

```
1
2 PERSONAJEX = 20
3 PERSONAJEY = 460
4 ARBUSTOX = -100
5 ARBUSTOY = 450
6 METAX = 620
7 METAY = 470
8
```



Estas Variables definen la posición inicial de nuestras imágenes que cargamos anteriormente, puedes observar que pertenecen a las coordenadas X y Y



```
1
2 NUBE1_DIM = [100,50,200,100]
3 NUBE2_DIM = [300,100,200,100]
4
```

Estos objetos son rectángulos que nos ayudaran a dibujar algunas nubes, 2 para ser exactos, estos rectángulos definen la posición inicial x,y así como el ancho y alto de la nube(2do y 3er parametro)

Ahora creamos otro similar solo que cambiara el tamaño del texto a 60, el mensaje y la posición donde se mostrara, este sera el mensaje utilizado para mostrar al usuario cuando haya ganado el juego

```
1
2 pygame.mixer.music.load("Molinos_de_viento.mp3")
3 pygame.mixer.music.play(-1,0.0)
4
```

Ahora como todo juego no podia faltar el sonido, para esto nosotros utilizaremos una canción de sonido de fondo, utilizamos los metodos de pygame music, cargamos primero escribiendo el nombre de la canción, puede ser wav o mp3, recuerda que estos recursos externos tienen que estar en la misma carpeta que nuestro archivo fuente, establecemos -1 para indicar que una vez que termine vuelva a empezar.

Todo juego en pygame necesita un ciclo principal, que se estara repitiendo a lo largo de la ejecución de nuestro juego, recuerda que en python la indentación juega un papel muy importante, nuestro ciclo principal iniciara así

```
1
2 while True:
3     PANTALLA.fill(CIELO)
4     pygame.draw.circle(PANTALLA,SOL,(700,100),120,0)
5     pygame.draw.ellipse(PANTALLA,NUBE,NUBE1_DIM,0)
6     pygame.draw.ellipse(PANTALLA,NUBE,NUBE2_DIM,0)
7
```

```
1
2 cumX = 0
3 cumY = 0
4
```

Variables auxiliares utilizadas para mover nuestro personaje, las utilizaremos mas tarde

```
1
2 fontObject = pygame.font.Font("StripeAttack.ttf",40)
3 textSurfaceObject = fontObject.render("Primer Juego en PyGame",True,ROJO,CIELO)
4 textRectObject = textSurfaceObject.get_rect()
5 textRectObject.center = (200,30)
6
```

Ahora para mostrar un mensaje en nuestro juego con un tipo de letra, tamaño y color, vamos a crear un objeto con el tipo de letra (que debe estar en la misma carpeta que el archivo de codigo fuente) y el tamaño de fuente, escribimos el texto que sera mostrado asi como otros parametros que son: True, antialiasing para mejorar la calidad del texto, ROJO, el color principal del texto, CIELO, color de fondo de nuestro rectangulo que contiene el texto. Establecemos el centro de nuestro texto

```
1
2 fontObject2 = pygame.font.Font("StripeAttack.ttf",60)
3 textSurfaceObject2 = fontObject2.render("Has Ganado el Juego",True,ROJO,CIELO)
4 textRectObject2 = textSurfaceObject.get_rect()
5 textRectObject2.center = (400,300)
6
```

Ahora creamos otro similar solo que cambiara el tamaño del texto a 60, el mensaje y la posicion donde se mostrara, este sera el mensaje utilizado para mostrar al usuario cuando haya ganado el juego

```
1
2 pygame.mixer.music.load("Molinos_de_viento.mp3")
3 pygame.mixer.music.play(-1,0.0)
4
```

Ahora como todo juego no podia faltar el sonido, para esto nosotros utilizaremos una cancion de sonido de fondo, utilizamos los metodos de pygame music, cargamos primero escribiendo el nombre de la cancion, puede ser wav o mp3, recuerda que estos recursos externos tienen que estar en la misma carpeta que nuestro archivo fuente, establecemos -1 para indicar que unavez que termine vuelva a empezar.

Todo juego en pygame necesita un ciclo principal, que se estara repitiendo a lo largo de la ejecucion de nuestro juego, recuerda que en python la indentacion juega un papel muy importante, nuestro ciclo principal iniciara asi

```
1
2 while True:
3     PANTALLA.fill(CIELO)
4     pygame.draw.circle(PANTALLA,SOL,(700,100),120,0)
5     pygame.draw.ellipse(PANTALLA,NUBE,NUBE1_DIM,0)
6     pygame.draw.ellipse(PANTALLA,NUBE,NUBE2_DIM,0)
7
```

Empezamos nuestro ciclo principal, utilizamos nuestro objeto pantalla que definimos antes para rellenar la pantalla de un color azul cielo que tambien hemos realizado ya, haremos uso de algunos metodos de pygame para dibujar figuras geometricas como el ellipse y el circulo, los cuales ocuparemos para crear un sol muy simple y un par de nubes, a estos metodos le pasaremos nuestro objeto pantalla, coordenadas y tamaños de figuras, en el caso de los ellipses pasaremos rectangulos.

```

1
2 PANTALLA.blit(ARBUSTO,(ARBUSTOX,ARBUSTOY))
3 PANTALLA.blit(ARBUSTO,(ARBUSTOX+200,ARBUSTOY))
4 PANTALLA.blit(ARBUSTO,(ARBUSTOX+340,ARBUSTOY))
5 PANTALLA.blit(ARBUSTO,(ARBUSTOX+500,ARBUSTOY))
6 PANTALLA.blit(ARBUSTO,(ARBUSTOX+660,ARBUSTOY))
7 PANTALLA.blit(META,(METAX,METAY))
8 PANTALLA.blit(PERSONAJE,(PERSONAJEX,PERSONAJEY))
9 PANTALLA.blit(textSurfaceObject,textRectObject)
10

```

Ahora mediante el metodo blit de nuestro objeto pantalla agregaremos elementos a nuestra pantalla, agregaremos algunos arbustos con el mismo objeto arbusto pero diferentes coordenadas, tambien agregaremos la imagen de meta mediante su objeto y sus coordenadas, agregamos nuestro personaje y por ultimos nuestro objeto de texto, el primero que creamos no el segundo.

```

1
2 PERSONAJEX = PERSONAJEX + aumX
3 PERSONAJEX = PERSONAJEX + aumX
4

```

En este par de lineas sumamos a las coordenadas de X de personaje, es decir a las de derecha/izquierda, el contenido de la variable aumX que mas adelante modificaremos segun si presionamos la tecla de derecha o izquierda de nuestro teclado, esto hara que nuestro personaje se mueva de izquierda a derecha, al modificar la posicion X

```

1
2 if PERSONAJEX >= 800:
3     PANTALLA.blit(textSurfaceObject2,textRectObject2)
4

```

en este conficional comprobamos si las coordenadas de X de nuestro personaje sobrepasan los 800, tamaño que le dimos de largo a nuestro juego, si es asi significa que nuestro personaje se ha movido lo suficiente a la derecha que ya no es visible en la pantalla, lo que significa que ha cruzado la meta, entonces se mostrara el segundo mensaje de texto que dice que el jugador ha ganado

```

1
2 for event in pygame.event.get():
3     if event.type == QUIT:
4         pygame.mixer.music.stop()
5         pygame.quit()
6         sys.exit()
7     elif event.type == pygame.KEYDOWN:
8         if event.key == pygame.K_RIGHT:
9             aumX = 5
10        if event.key == pygame.K_LEFT:
11            aumX = -5
12        elif event.type == pygame.KEYUP:
13            if event.key == pygame.K_RIGHT:
14                aumX = 0
15            if event.key == pygame.K_LEFT:
16                aumX = 0
17

```

Aquí trataremos los eventos de nuestro juego, en esta ocasión solo ocuparemos el básico de `QUIT` para salir correctamente, y los de `KEYDOWN` Y `KEYUP`, para saber cuando se presione y se libere una tecla, en particular la tecla derecha (`K_RIGHT`) Y izquierda (`K_LEFT`) que son las que ocupamos para movernos de izquierda a derecha, al detectar este evento estableceremos el aumento en positivo 5 para derecha o negativo 5 para izquierda y cuando se suelte la tecla estableceremos el aumento en 0, esto es, sin movimiento.

```
1  
2 pygame.display.update()  
3
```

Con esta instrucción se actualiza la pantalla de nuestro juego

```
1  
2 frecuencia.tick(FPS)  
3
```

Establecemos la frecuencia de nuestro juego que fue de 30

Se intentó plantear el cuestionario de una manera similar a la del juego anterior, donde se introduciría un archivo de texto con las preguntas y las posibles respuestas, y dependiendo de la respuesta, como ya se ha mencionado, saldrá el vídeo. Sin embargo no se consiguió hacer, ya que resultó muy complicado comprender esta librería de Python.

Dado que la base de programación de la que disponemos no era suficiente para elaborar el programa interactivo utilizando Python y el tiempo que restaba para la culminación del proyecto, se tuvo que descartar la idea de elaborar la verificación del código como se había planeado al principio.

#### 6.2.4 SIENA

Como se mencionó anteriormente, elaborar el cuestionario con un lenguaje de programación fue una tarea que desafortunadamente no se pudo cumplir.

Así que se optó por otra alternativa que consiste en crear el cuestionario en un entorno web, donde los alumnos puedan acceder desde cualquier lugar, puedan prepararse la teoría, elaborar trabajos en grupo marcados por el profesor, entre otras cosas.

Para ello se escoge el programa interactivo creado por la Universidad de la Laguna, SIENA (Sistema Integrado de Enseñanza Aprendizaje).

Antes el cuestionario constaría de un máximo de 50 preguntas, en esta ocasión el cuestionario constará con más de 60 preguntas, donde se plantean las distintas situaciones por las que debe pasar la estación de acarreo de piezas, se muestran por medio de un enlace los vídeos de la simulación en 3D que se crearon en Inventor y demás.

En el siguiente capítulo se podrá ver cómo se elaboran las preguntas en este entorno Web, cómo insertar los archivos multimedia (el principal inconveniente de antes). También se muestra el mapa conceptual, necesario para SIENA, donde se ven los temas a tratar para el proyecto.

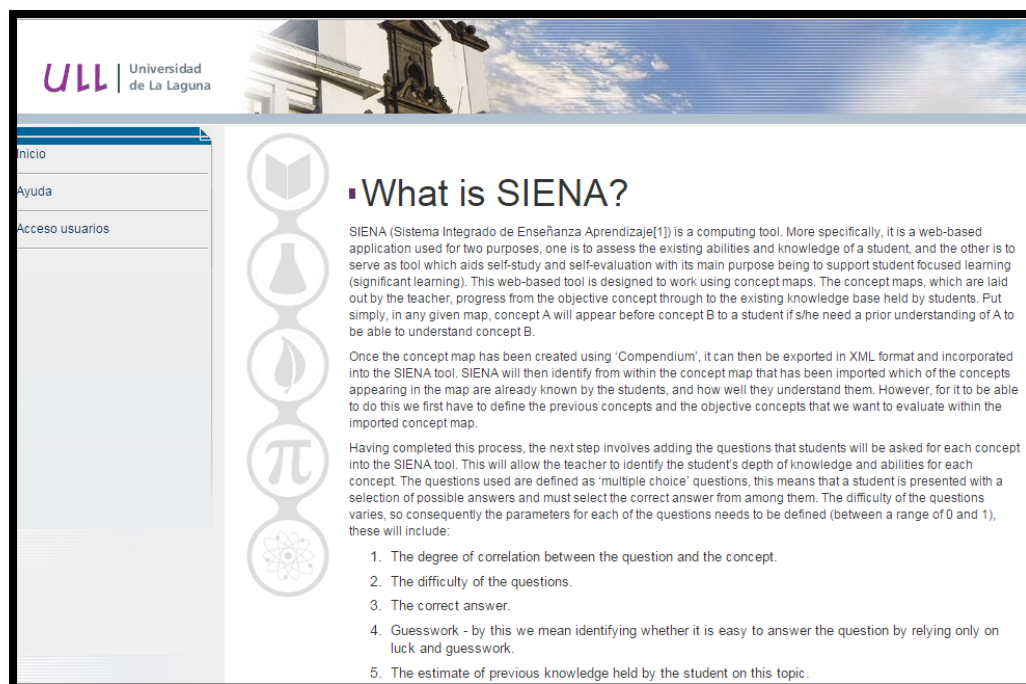


Figura 6.14 Entorno de SIENA

## Capítulo 7. Software de aprendizaje para la enseñanza en la ingeniería.

### 7.1 Objetivo

El objetivo de este programa es mejorar los conceptos adquiridos en dicha asignatura como la programación en Step 7 Microwin, motivar a los alumnos para que aprendan de forma autónoma la programación de CoDeSyS para el lenguaje de diagrama de contactos y para el de listas de instrucciones. Esto permitirá a los estudiantes hacer una comparación de este programa con Microwin. También se tratará el tema del estándar IEC 61131-3.

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

### 7.2 ¿Qué es SIENA?

SIENA (Sistema Integrado de Enseñanza Aprendizaje) es una aplicación creada por la Universidad de La Laguna, basada en web que se utiliza para evaluar las habilidades y conocimientos existentes de un estudiante, y para servir como herramienta que ayuda al autoaprendizaje y la autoevaluación, apoyándose de mapas conceptuales de la asignatura para la que se utiliza esta aplicación. Se basa en dos elementos clave: los mapas conceptuales y los tests adaptativos

#### 7.2.1 ¿Cómo funciona?

SIENA está diseñado para trabajar con mapas conceptuales. Los mapas conceptuales se utilizan para organizar y representar los conocimientos de un tema concreto de manera gráfica. El uso de este tipo de herramientas mejora la comprensión y retención de las ideas, ayudan a la memorización de conceptos y relaciones, permiten personalizar el aprendizaje, la compartición de conocimiento y refuerza las habilidades del aprendizaje.

Éstos se elaboran utilizando el software **compendiumLD**. Una vez que se ha elaborado el mapa conceptual, éste se exporta en formato XML para insertarlo en la herramienta SIENA.

SIENA identifica dentro del mapa conceptual cuáles de los conceptos que aparecen son conocidos por los estudiantes. Sin embargo, para que sea capaz de hacer esto, primero se tienen que definir los conceptos anteriores y los conceptos objetivos que queremos evaluar dentro del mapa conceptual importado.

Luego de realizar este proceso, se realizan las preguntas sobre los conceptos de los que se va a evaluar a los estudiantes, esto permitirá al maestro identificar los conocimientos y habilidades de los alumnos para cada uno de los temas tratados en el mapa conceptual.

Las preguntas utilizadas se definen como preguntas "opción múltiple", lo que significa que al estudiante se le presenta una selección de respuestas posibles y debe seleccionar la respuesta correcta. La dificultad de las preguntas varía, por lo que en consecuencia, los parámetros para cada una de las preguntas necesitan ser definidos (entre un rango de 0 y 1), en éstos se incluyen:

- El grado de correlación entre la pregunta y el concepto.
- La dificultad de las preguntas.
- La respuesta correcta.
- Conjeturas - Con esto queremos decir identificar si es fácil responder a la pregunta por confiar sólo en la suerte y conjeturas.
- La estimación de los conocimientos previos llevados a cabo por el estudiante sobre este tema.
- Tiempo de respuesta (en segundos) que se les da a los estudiantes para responder a cada pregunta.

La razón de esto es que SIENA incluye una prueba de adaptación que se basa en la búsqueda de redes bayesianas (que modelan un fenómeno mediante un conjunto de variables y las relaciones de dependencia entre ellas). Es por ello que todos estos parámetros tienen que estar en su lugar para que SIENA sea capaz de estimar correctamente la capacidad y la comprensión de un concepto dado, esto depende de los resultados de la prueba de cada estudiante.

Para poder hacer esta estimación, la prueba se le presenta a los alumnos con preguntas cada vez más difíciles, es decir, si el estudiante responde correctamente, el sistema aumentará el nivel de dificultad de la pregunta siguiente; sin embargo, si en un momento dado el estudiante no responde a una pregunta correctamente, entonces el nivel de dificultad se reducirá en la siguiente pregunta.

Además de esta función, el sistema también incluye un mecanismo de tope que es útil cuando ya no es posible obtener una estimación adicional de que tan bien se entiende un concepto, o porque la herramienta se ha quedado sin preguntas.

En la autoevaluación, SIENA solicita al maestro diseñar una tarea para que el estudiante la complete. Las tareas se definen como partes del mapa conceptual que se consideran consistentes con un nodo objetivo en particular, que será un nodo intermedio, que el estudiante espera alcanzar durante la autoevaluación, así como un nodo inicial que coincide con el último nodo del mapa que se utilizó como el nodo objetivo en una etapa anterior de la auto-evaluación.

Una tarea completada coincidirá con la totalidad del mapa conceptual. Las tareas se fijan en los casos en que un mapa conceptual es muy grande y el maestro decide que es preferible que el estudiante progrese en etapas. El proceso comienza examinando los conceptos anteriores que se definen en el mapa y luego comienza a evaluar los conceptos progresivamente en el mapa, pero sólo cuando el estudiante pasa un concepto dado con una marca de por lo menos 0,5. Cuando un concepto no se ha superado con éxito, el sistema no continúa evaluando al estudiante a lo largo de la misma rama del mapa, ya que se supone que si el concepto no ha sido aprobado, tampoco lo será el siguiente.

Las pruebas pueden ser realizadas por un estudiante en particular, o por un grupo de estudiantes que trabajan juntos en una tarea de colaboración en línea. En este último, los alumnos se comunican mediante un chat de mensajería instantánea donde pueden discutir las posibles respuestas de las preguntas de la prueba.

Para los estudiantes individuales, SIENA produce la estimación de los conocimientos del estudiante mediante el uso de las respuestas a las preguntas que se presentan por la prueba.

En el caso de las pruebas en línea de colaboración, SIENA utiliza las respuestas a las preguntas para estimar el conocimiento del grupo. Además de predecir el conocimiento del grupo, también presenta el número de mensajes que se envían por cada estudiante y también el contenido de dichos mensajes.

Con esta información el profesor es capaz de evaluar objetivamente la contribución de cada estudiante durante la tarea de colaboración.

SIENA también incluye contenido para cada nodo del mapa conceptual, que ofrece la posibilidad de combinar el auto-estudio con la auto-evaluación. En realidad, el sistema está preparado para incorporar dos tipos de contenido: contenido que el alumno ha de estudiar la



primera vez que se ocupan de un concepto, o repetir el contenido que debe ser estudiado cuando un estudiante ha fallado una prueba en ese concepto. [16]

### 7.2.2 Registrarse en SIENA

Lo primero que hay que hacer es registrarse en SIENA. Dentro de este programa se distinguen tres tipos de usuarios:

- Administrador:  
El administrador es el que se encarga de agregar nuevos temas. Éste debe activar a los usuarios (alumnos y profesores) que se han registrado. También puede ver, editar y eliminar preguntas guardadas y nodos, para todos los mapas guardados, desde la lista de preguntas hasta la lista de nodos respectivamente.
- Maestro:  
El maestro puede ver a los alumnos matriculados, nodos asociados, preguntas asociadas, y la presentación visual de un mapa conceptual. También puede asignar una tarea / ejercicio para los estudiantes registrados y ver la lista completa sujetos.
- Estudiante:  
Una vez que el estudiante se ha registrado y ha sido activado por el administrador, el profesor de la asignatura asignará una tarea para que el estudiante la complete. El estudiante puede tomar una prueba de que se ha establecido sólo para ellos, el acceso a la materia y la tarea para ese tema. También puede acceder al contenido de los nodos de la tarea primero y luego tomar la prueba.

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

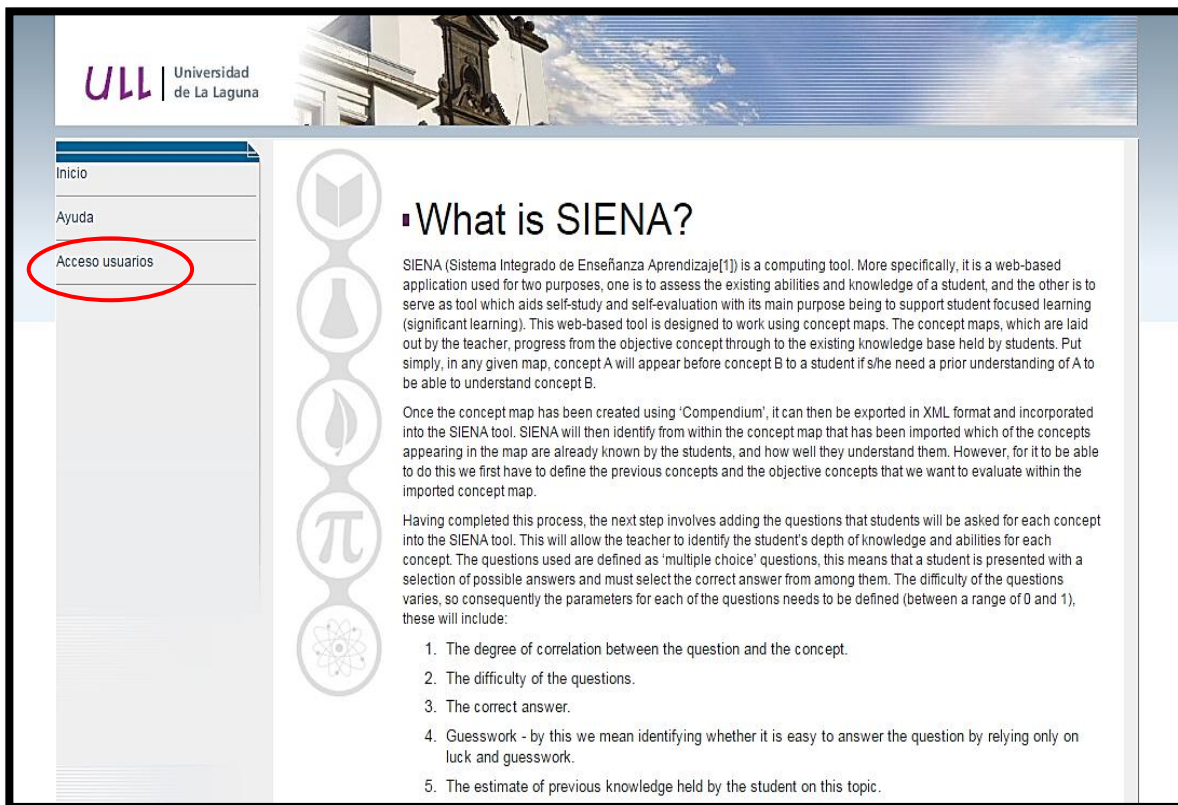


Figura 7.1 Siena social

Para registrarse hay que localizar la parte de la pantalla que dice “Acceso Usuario”:

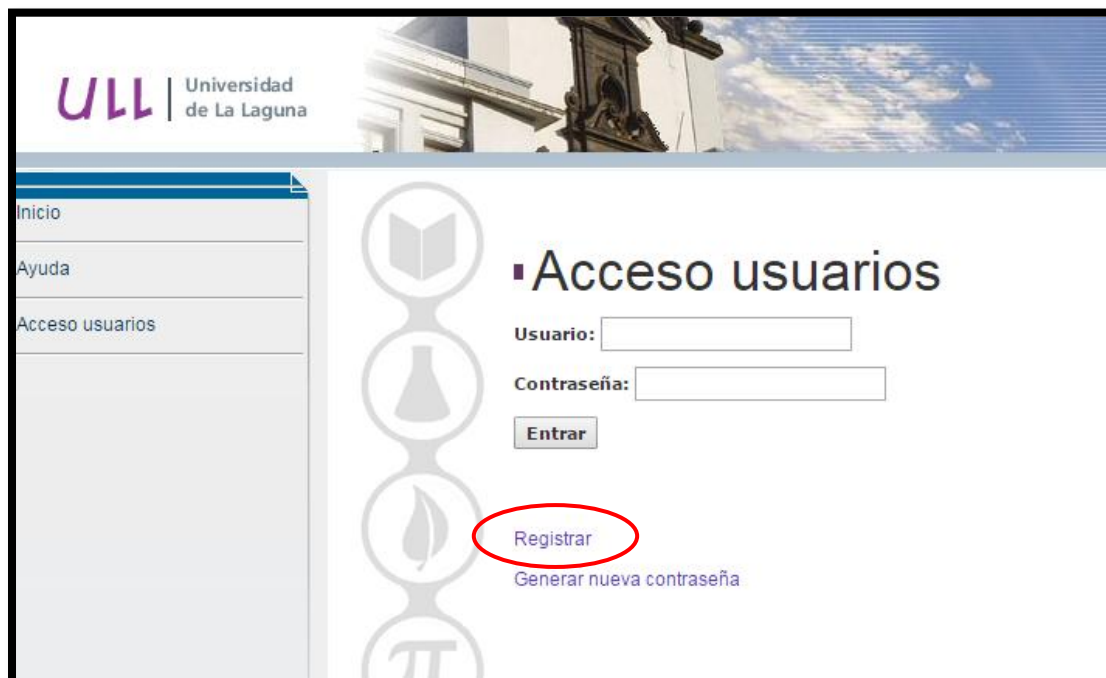
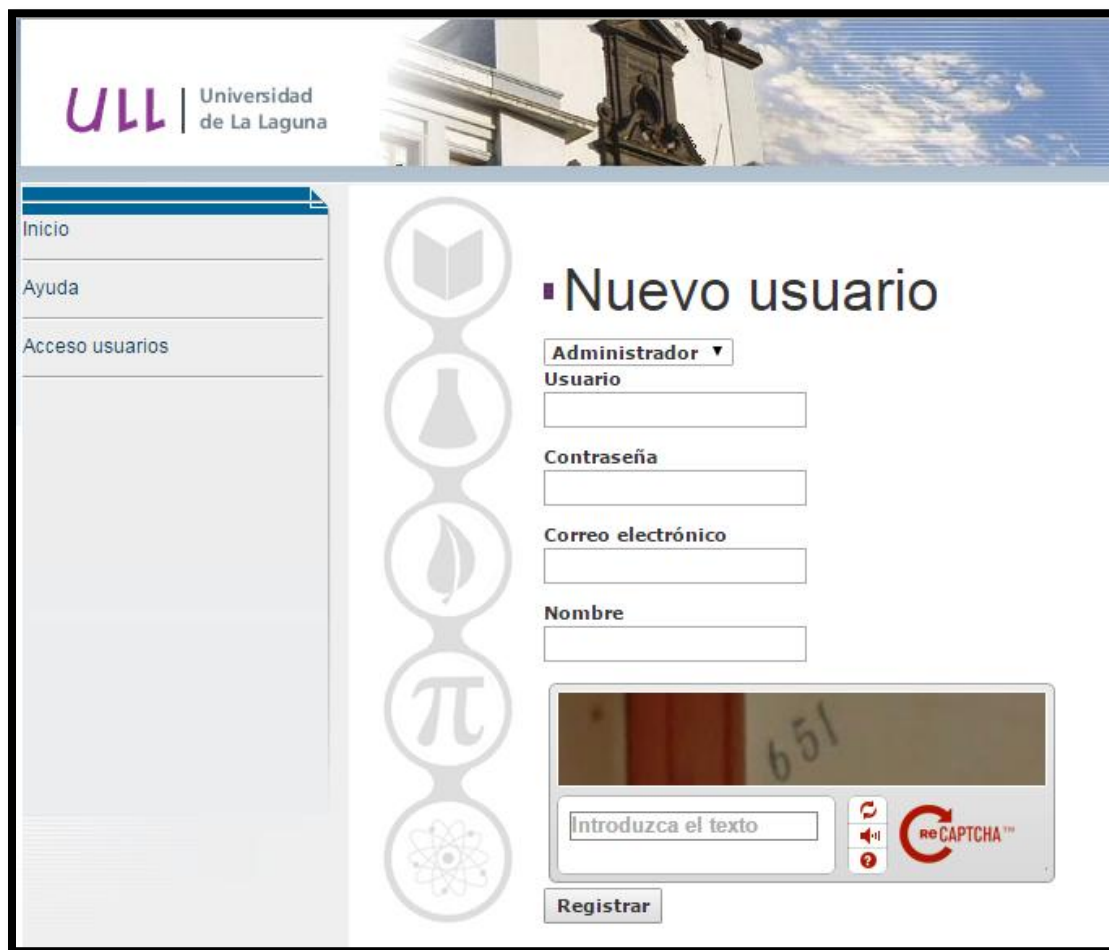


Figura 7.2 Acceso Usuarios

Se selecciona “Registrar” para un nuevo usuario:



The screenshot shows the 'Nuevo usuario' registration page on the ULL website. The page features a navigation menu on the left with links for 'Inicio', 'Ayuda', and 'Acceso usuarios'. The main content area is titled 'Nuevo usuario' and includes a dropdown menu for 'Administrador' (set to 'Administrador'), and input fields for 'Usuario', 'Contraseña', 'Correo electrónico', and 'Nombre'. A reCAPTCHA verification box is present, showing a distorted image of the number '651' and a 'ReCAPTCHA' logo. A 'Registrar' button is located at the bottom of the form.

Figura 7. 3 Nuevo Usuario

Para el usuario se escribe el nombre, sin espacios en blanco, es decir, un identificador. Se escribe la contraseña en los dos cuadros de texto que aparecen a continuación. Y finalmente se puede escribir tu nombre completo, incluyendo los espacios, en el último cuadro de texto.

Se puede solicitar el registro en cualquiera de los tipos de usuario, pero hasta que alguno de los administradores autorice el registro no es posible trabajar en SIENA.

Como el objetivo es poder elaborar el cuestionario y modificarlo, es necesario registrarse como maestro. Para poder probar si el cuestionario funciona correctamente también se crea una cuenta como alumno.

- **SIENA como maestro:**

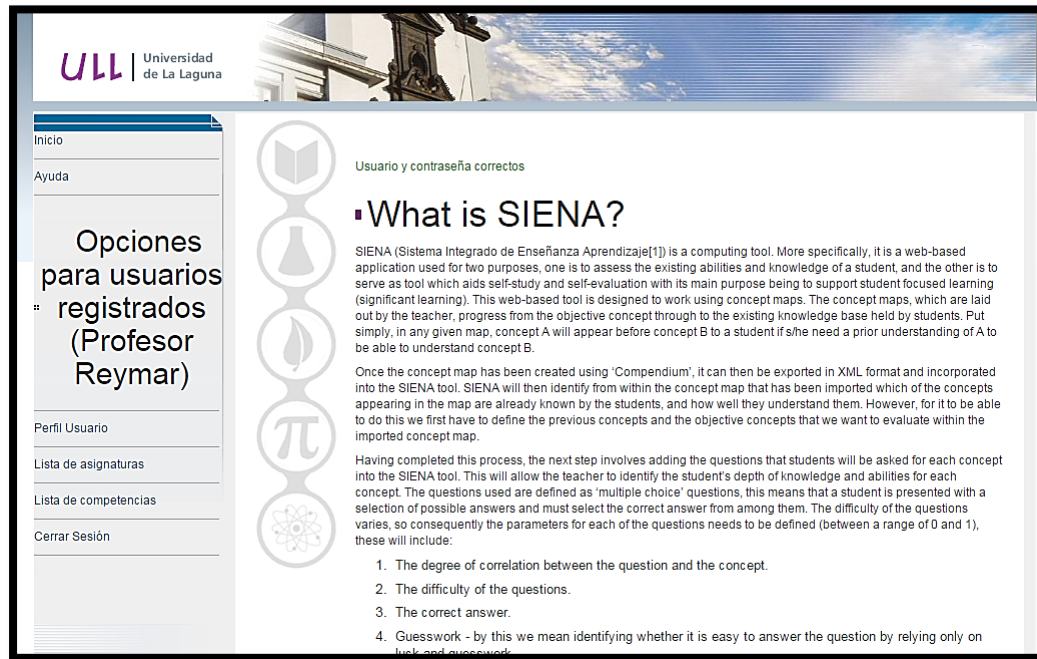


Figura 7.4 Profesor en SIENA

El menú de profesor es el siguiente:

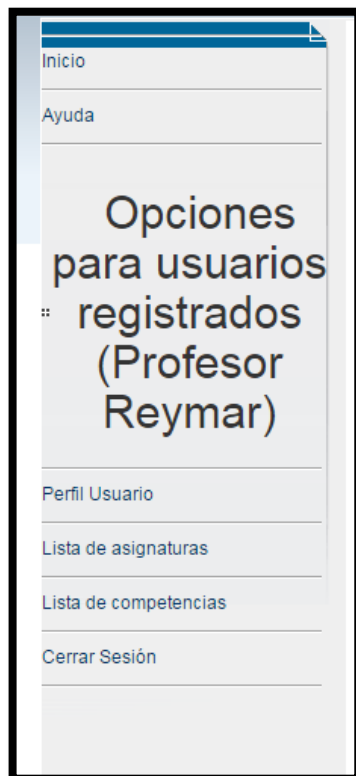


Figura 7.5 Menú para los profesores

En “Perfil de Usuario” se puede modificar el perfil del profesor.

En “Lista de asignaturas” se pueden ver las asignaturas creadas por los profesores para los alumnos. En este caso estará disponible la asignatura para la que se elabora el cuestionario, Automatización Industrial Avanzada. En “Lista de competencia” se añaden las competencias que se quiere desarrollar para los alumnos de la asignatura.

- **SIENA como alumno:**

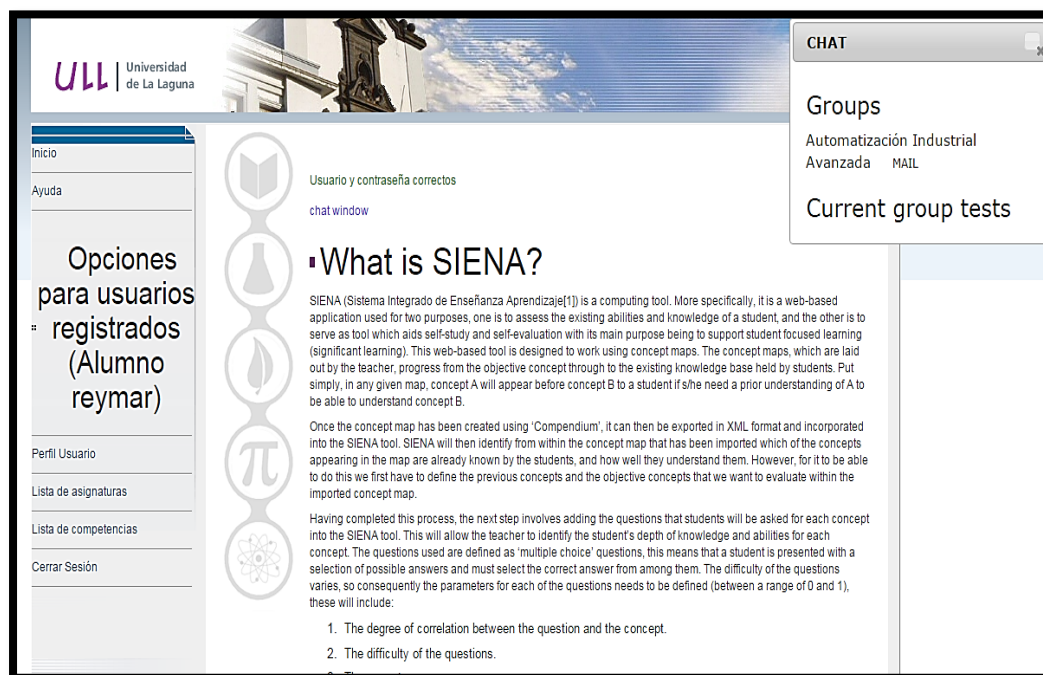


Figura 7.6 Menú para los alumnos

En este caso se puede ver que el menú de la izquierda es el mismo que el que se vio para los profesores, sin embargo las atribuciones de los alumnos no serán las mismas. Mientras los maestros pueden ver, editar y borrar cualquier tarea, cuestionarios, mapa conceptual entre otras cosas, el alumno accede a las asignaturas para realizar los trabajos marcados por el maestro, y los cuestionarios, sin poder modificarlos. También se puede ver la ventana del chat que se mencionó anteriormente, donde los alumnos pueden comunicarse entre ellos cuando se trate de un trabajo en grupo.

### 7.3 Elaboración del cuestionario

Una vez registrados en SIENA se procede a la elaboración del cuestionario para la asignatura Automatización Industrial Avanzada, como ya se ha mencionado en los capítulos anteriores.

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.

El mapa de la asignatura consta de un conjunto de nodos formados por un conjunto de conocimientos, habilidades, estrategias o destrezas que queremos desarrollar en el alumno. Preparar una asignatura (tema) para SIENA significa establecer un mapa conceptual jerárquico. Es decir, un mapa cuyos nodos superiores son los más complejos, y su logro o desempeño por parte del alumno, depende del logro en los nodos más simples e inferiores en el rango del mapa conceptual.

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.

El mapa conceptual se elabora con el programa CompendiumLD.

### 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, maestros y otras personas involucradas en la educación para ayudarles a expresar sus ideas y trazas la secuencia de diseño o de aprendizaje.

Esta herramienta viene con configuraciones predefinidas para los iconos, algunos genéricos y otros específicos de diseño de aprendizaje. [17]

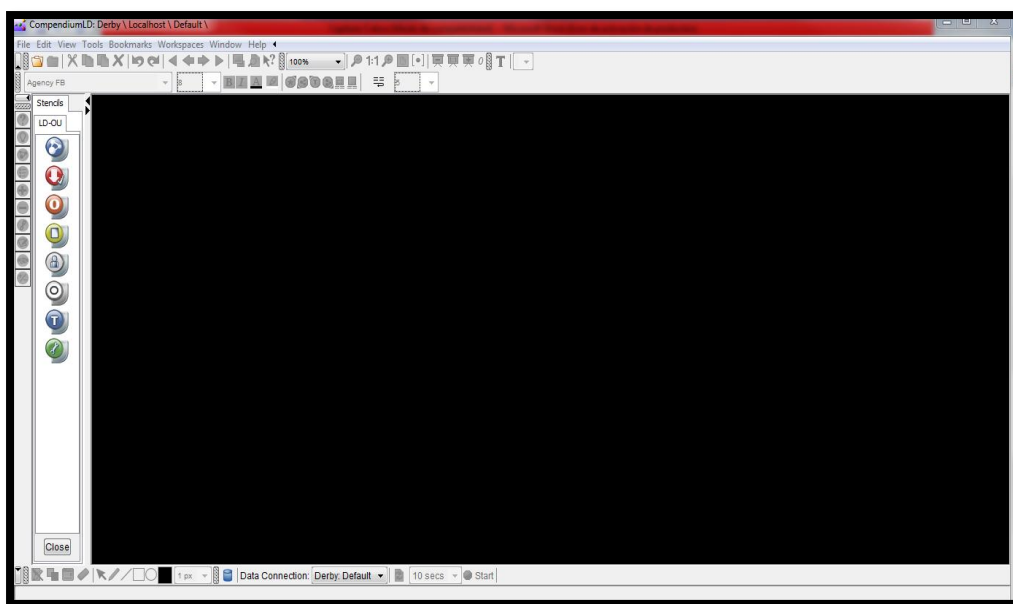


Figura 7.7 Interfaz de CompendiumLD

Para elaborar el mapa primero hay que crear un nuevo proyecto, esto se hace en file → new...

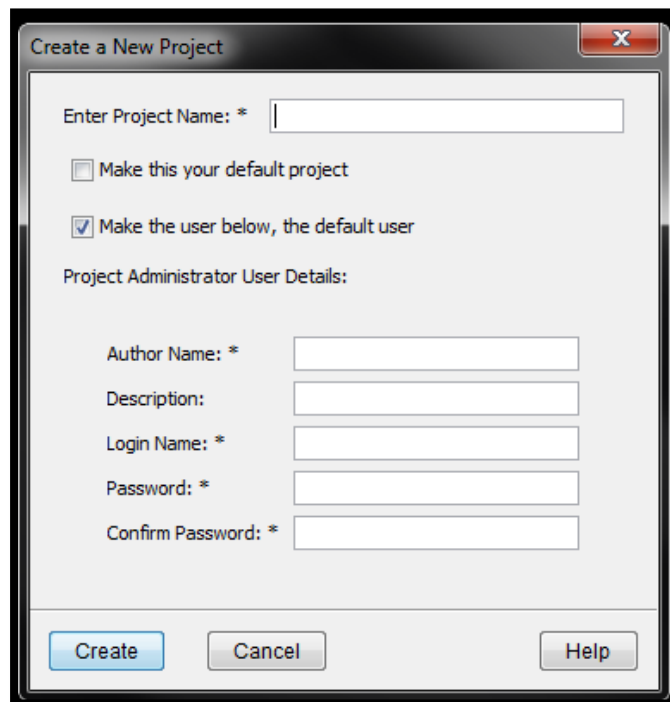


Figura 7. 8 Nuevo proyecto

En esta ventana se especifica el nombre del proyecto, nombre del autor, nombre de usuario y la contraseña. Una vez que se ha creado el proyecto aparece la siguiente ventana:

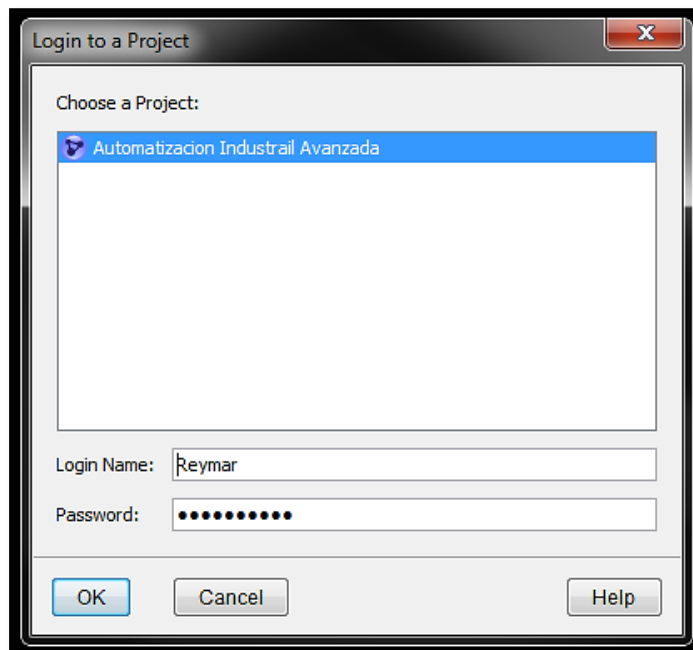


Figura 7. 9 Acceso al proyecto

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.

Para crear 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.

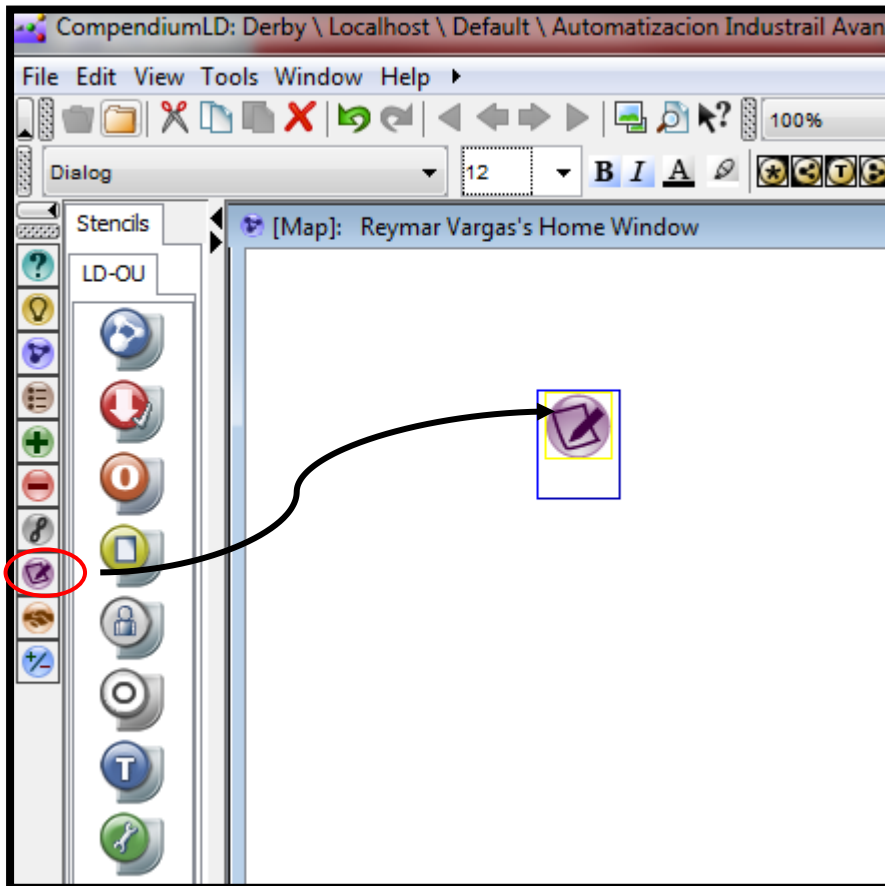


Figura 7.10 Creación de nodos

Los nodos que dependan de otro se unen haciendo click-derecho sobre el primero y sin soltarlo, se desliza hasta el segundo nodo.

Para añadir algún tipo de comentario entre los nodos, sólo hay que pulsar sobre la línea a una distancia media distancia (aproximada) de ambos nodos. Se activará un cuadro de texto que permite escribir la relación entre ambos nodos. Aunque dicho sea de paso, SIENA no tiene en cuenta los textos que hay en las líneas de unión del mapa.

Dicho esto, el mapa queda de la siguiente manera:



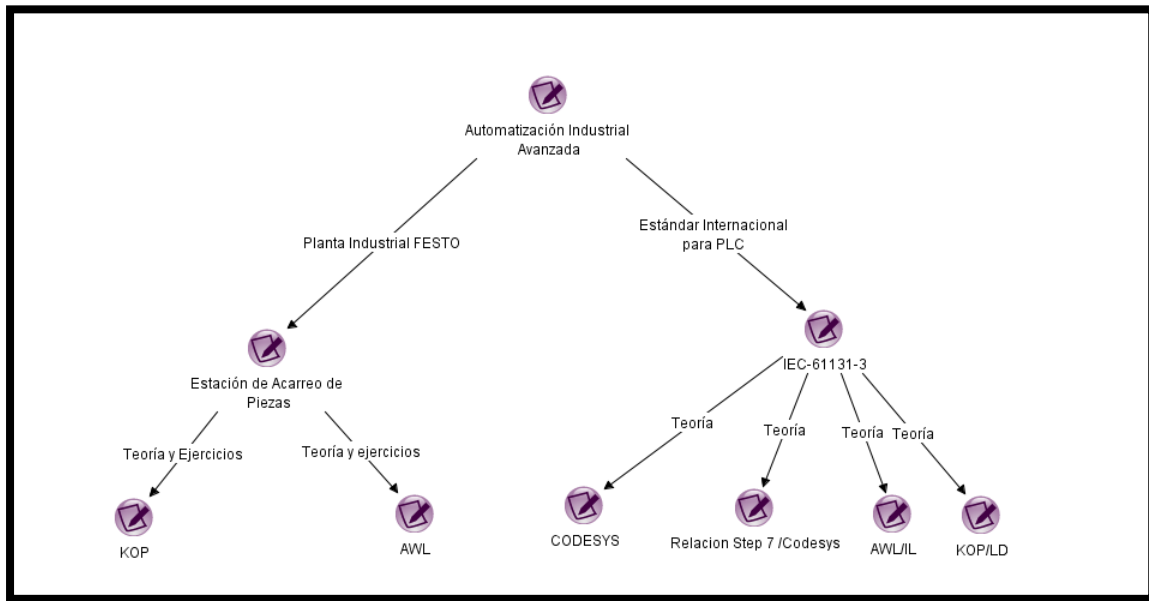


Figura 7.11 Mapa Conceptual para SIENA

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

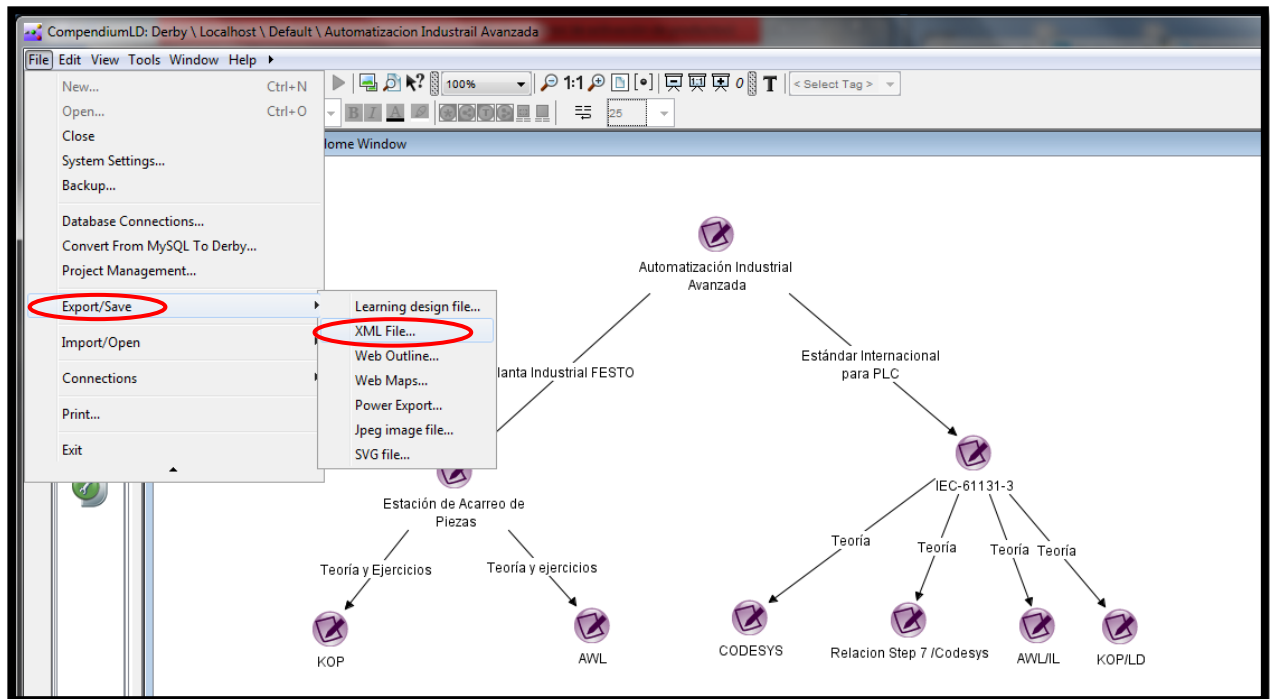


Figura 7.12 Exportar a XML

## SIENA

Una vez elaborado el mapa conceptual, se procede a la elaboración del cuestionario. Para ello hay que acceder a SIENA como profesor, como se mencionó anteriormente. Luego hay que matricularse y acceder a la asignatura para la que se realiza el cuestionario, en este caso Automatización Industrial Avanzada.

Se selecciona “Lista de Asignaturas” y a continuación aparecerá el siguiente recuadro:

Nombre	Ver	State	
<input type="text"/>			
Automatización Industrial Avanzada	<a href="#">Ver</a>	<a href="#">Desmatricular</a>	

Figura 7.13 Lista de asignaturas

Para acceder a la asignatura hay que pulsar la opción “ver”. Dentro de ésta aparece el siguiente menú:

**Nombre:** Automatización Industrial Avanzada

- [Lista de Grupos](#)
- [Lista de alumnos matriculados en Automatización Industrial Avanzada](#)
- [Lista de nodos de la asignatura](#)
- [Lista de preguntas de la asignatura](#)
- [Mapa de la asignatura \(imagen\) | \[Generar imagen\]\(#\)](#)
- [Lista de asignaturas](#)

Figura 7.14 Menú de la asignatura

En la sección “Lista de Grupos” se puede crear los grupos de alumnos, ver los grupos para comprobar los integrantes del mismo. Editar para añadir nuevos alumnos al grupo, o eliminar un grupo que se ha creado previamente.

En “Lista de alumnos” están los alumnos matriculados en la asignatura.

En la “Lista de nodos” se encuentran los conceptos que se importan desde el mapa conceptual.

En la “Lista de preguntas” se puede ver todas las preguntas elaboradas para la asignatura.

En “Mapa de la asignatura” está la imagen del mapa conceptual que se exporta a SIENA desde CompendiumLD. Primero hay que generar la imagen.

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



Figura 7.15 Importar mapa a SIENA

Una vez importado, se procede a generar la imagen del mapa:

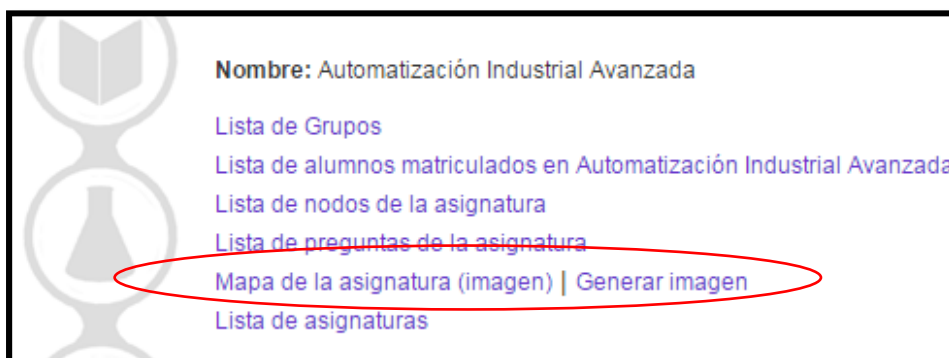


Figura 7. 16 Generar imagen del mapa

Cuando se genera la imagen aparece el siguiente mapa:

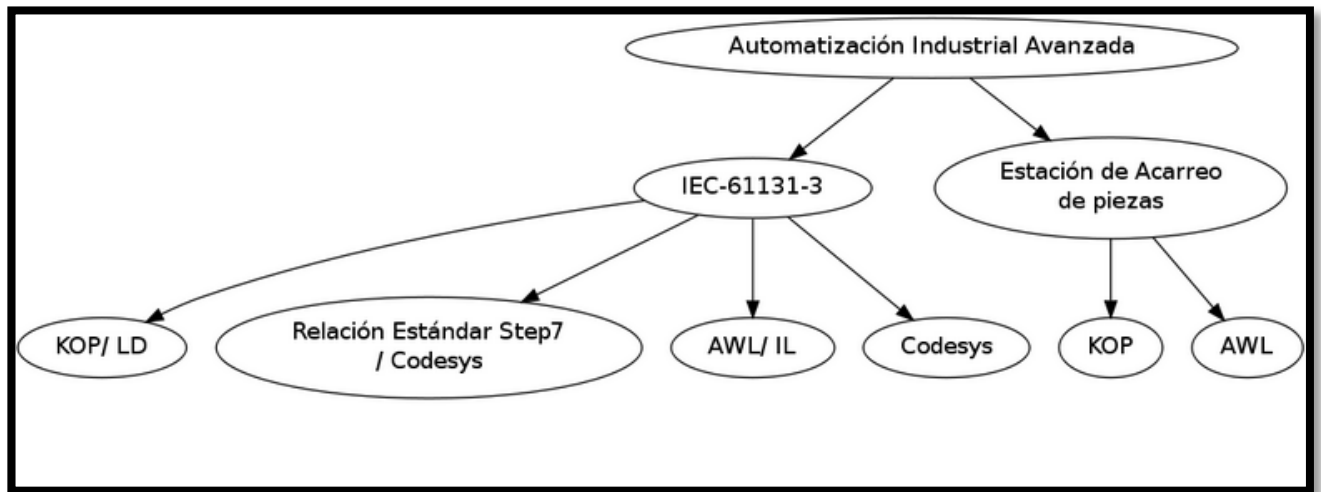


Figura 7.17 Mapa de la asignatura

Luego de generar la imagen, 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.

**Nueva pregunta**

**Contenido**

**Tiempo de respuesta**

**Respuesta correcta**

**Dificultad**

**Adivinanza**

**Imagen**  
 Ningún archivo seleccionado

**Keywords**

[Atrás](#)

Figura 7.18 Elaboración de la 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 correcta menos uno, es decir, si la pregunta correcta es la tercera en esta casilla se pondrá el número dos.

En “dificultad” se añade un número comprendido 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.

Siena permite la posibilidad de insertar un archivo a la pregunta.

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



**Contenido:** Indique el nombre del temporizador, que se muestra a continuación, utilizado en Step 7 y la instrucción con la que se accede a él, en el lenguaje AWL.

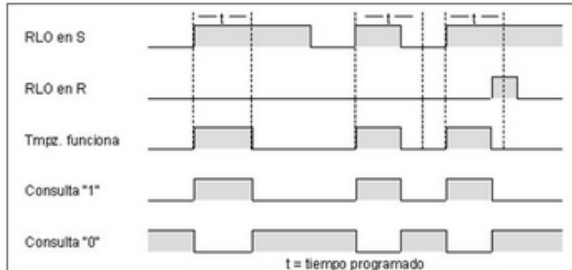
**Tiempo de respuesta:** 120

**Respuesta correcta:** 1

**Dificultad:** 0.5

**Adivinanza:** 0.25

**Imagen:**



**Keywords:** Propuestos:

**Creado por:** Reymar Andrea Vargas Iniesta

**Asignatura:** Automatización Industrial Avanzada

**Respuestas**

- Temporizador como retardo a la desconexión, instrucción "SA"
- Temporizador como impulso, instrucción "SI"
- Temporizador como retardo a la conexión con memoria, instrucción "SS"
- Ninguna de las anteriores

Figura 7.19 Pregunta creada en SIENA

Cuando se redacta una pregunta, ésta debe relacionarse con alguno de los nodos de la asignatura. En este caso la pregunta está relacionada con el lenguaje de programación AWL.

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.

## Asociar nuevo nodo a la pregunta

**Nodo:**

<input checked="" type="checkbox"/> <input type="checkbox"/>	Asignatura	Contenido	
	Automatización Industrial Avanzada ▾ ⊕	<input type="text"/>	Filtrar →
<input type="checkbox"/>	Automatización Industrial Avanzada	KOP/ LD	
<input type="checkbox"/>	Automatización Industrial Avanzada	KOP	
<input type="checkbox"/>	Automatización Industrial Avanzada	AWL	
<input type="checkbox"/>	Automatización Industrial Avanzada	Relación Estándar Step7 / Codesys	
<input type="checkbox"/>	Automatización Industrial Avanzada	Automatización Industrial Avanzada	
<input type="checkbox"/>	Automatización Industrial Avanzada	IEC-61131-3	
<input type="checkbox"/>	Automatización Industrial Avanzada	AWL/ IL	
<input type="checkbox"/>	Automatización Industrial Avanzada	Estación de Acarreo de piezas	
<input type="checkbox"/>	Automatización Industrial Avanzada	Codesys	

1-9 / 9

**Dependencia:**

Figura 7. 20 Nodos disponibles para la asignatura

Dentro de esta lista se puede ver la opción “Dependencia”. Esta relaciona el contenido de la pregunta con el nodo. El rango de dependencia es de 0 a1.

Una vez asociado el nodo a la pregunta y la dependencia entre ellos, aparecerá un recuadro en la parte inferior de la pregunta con la relación que se hizo anteriormente:

Nodos relacionados					
Asignatura	Nombre	Editar relación entre el nodo y la pregunta	Borrar relación entre el nodo y la pregunta	Editar nodo	Dependencia
Automatización Industrial Avanzada	AWL	<a href="#">Editar</a>	<a href="#">Borrar</a>	<a href="#">Editar</a>	1.0
					1-1 / 1

Figura 7.21 Nodos relacionados

Este proceso se debe seguir para cada una de las preguntas elaboradas.

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 elaborada, permitiendo ver, editar o borrar cualquiera de éstas y el usuario que crea la pregunta, este caso el usuario es Reymar.

Lista de preguntas de la asignatura					
Nombre	Usuario	Ver	Editar	Borrar	
<input type="text"/>	<input type="text"/>				
¿Qué tipo de contador se representa en la siguiente imagen?	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	
¿Qué operación lógica estamos empleando en el siguiente network?	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	
Las bobinas pueden ir precedidas y seguidas de contactos	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	
Las subrutinas sirven para estructurar un programa en bloques más pequeños, por lo que realizar las tareas de comprobación y mantenimiento del programa es mucho más complejo.	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	
En el software de programación Step 7 Microwin ¿qué se entiende por subrutina?	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	
Existen tres tipos de subrutinas: Llamadas múltiples, Anidadas y de Final múltiple	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	
La operación RET se utiliza como la ultima instrucción de una subrutina y devuelve esta al sitio correcto del programa de invocación.	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	
Identifica a qué tipo de temporizador pertenece el siguiente esquema.	Reymar	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Borrar</a>	

Figura 7.22 Lista de preguntas



Al final de la lista se puede ver el número preguntas que se han elaborado:



Figura 7.23 Preguntas elaboradas para el cuestionario

Cuando un alumno accede a SIENA y se matricula en una asignatura, se encuentra con las siguientes listas:

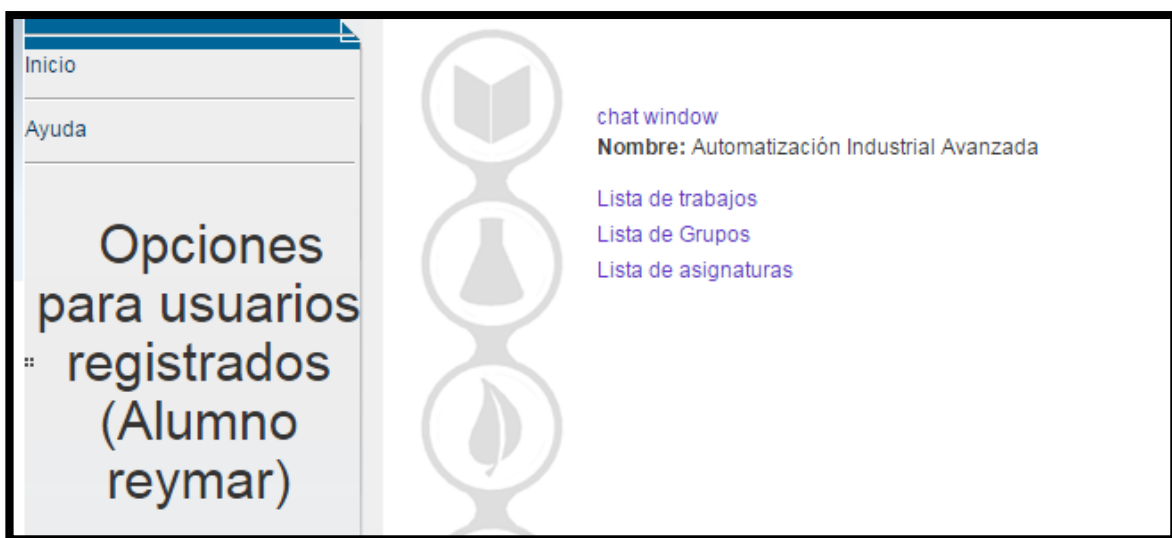


Figura 7. 24 Interfaz de SIENA para alumnos

En la Lista de trabajos el alumno puede ver los nodos relacionados con la asignatura y el contenido que debe leer. En caso de que el alumno todavía no haya realizado el cuestionario aparece la opción “contenidos” en el que se puede ver el material que el profesor añade para que se preparen antes de hacer el cuestionario, si por el contrario el alumno ya ha realizado el cuestionario y desea repetirlo, aparece la opción “contenidos (recuperación)”. También se puede acceder al cuestionario.

chat window

## Lista de trabajos de Reymar Vargas para Automatización Industrial Avanzada

Nodo	Conocimiento previo		
KOP/LD	0.0	TestContenidos(recuperación)	No PasadoVer tests
KOP	0.0	TestContenidos(recuperación)	No PasadoVer tests
AWL	0.0	TestContenidos(recuperación)	No PasadoVer tests
Relación Estándar Step7 / Codesys	0.0	TestContenidos	No Pasado
Automatización Industrial Avanzada		Contenidos	No Pasado
IEC-61131-3	0.0	Contenidos	No Pasado
AWL/IL	0.0	TestContenidos	No Pasado
Estación de Acarreo de piezas	0.0	Contenidos	No Pasado
Codesys	0.0	TestContenidos	No Pasado

Atrás

Figura 7.25 Lista de trabajos de la asignatura

En Lista de Grupos aparecen los grupos formados, por el profesor, que sean necesarios para un trabajo en concreto.

Inicio

Ayuda

Opciones para usuarios registrados (Alumno reymar)

chat window

## Lista de Grupos

Nombre	Lista de usuarios	Lista de trabajos
--------	-------------------	-------------------

Figura 7.26 Lista de grupos

En lista de asignatura el alumno tiene la posibilidad de ver todas las asignaturas a las que se puede matricular.

chat window

## Lista de asignaturas


Nombre	Ver	State	
<input type="text"/>			
Automatización Industrial Avanzada	Ver	Desmatricular	
CuerpoHumano		Matricular	
English MEDUTIC		Matricular	
Arquitectura Computadores Grado		Matricular	
pruebaArqAv		Matricular	
Arquitecturas Avanzadas		Matricular	
MEDUTIC		Matricular	
compen		Matricular	
siena_comp		Matricular	

Figura 7.27 Lista de asignaturas

La pregunta vista por los alumnos queda de la siguiente manera:

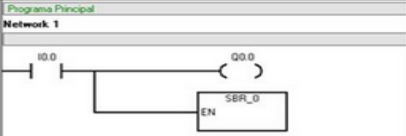
chat window

## Pregunta 1

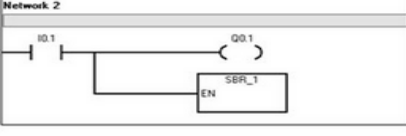
Identifica a qué tipo de temporizador pertenece cada esquema

Programa Principal

**Network 1**

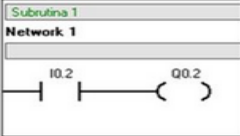


**Network 2**



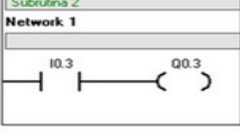
Subrutina 1

**Network 1**



Subrutina 2

**Network 1**



21 Segundos restantes

1. R/ Subrutina de Llamada múltiple
2. R/ Subrutina de Final múltiple
3. R/ Subrutina Anidada

Enviar

Figura 7.28 Pregunta del cuestionario

## 7.4 Aplicación de Inventor a SIENA.

Algunas de las preguntas que se elaboran en SIENA incluyen archivos multimedia como imágenes y vídeos. En la pregunta anteriormente citada se puede observar que la herramienta permite insertar imágenes. Sin embargo, no es posible adjuntar vídeos a SIENA, es por ello que se decidió crear un canal en Youtube, un sitio web en el cual los usuarios pueden subir y compartir vídeos.

En este canal se comparten los videos de la simulación creados en Inventor para el cuestionario.

En el caso en el que sea necesario que el alumno trabaje apoyándose de vídeos, se les dejará el enlace al que tienen que ir para poder verlos y a continuación pueda responder a la pregunta planteada.

El canal se llama **TFG Electrónica**, en él se encuentran los videos de la simulación realizados en inventor.



Figura 7.29 Canal de Youtube

A continuación se muestra un ejemplo en el que el alumno se debe dirigir al enlace para responder a la pregunta que se le plantea:

**Contenido:** En el siguiente enlace se puede observar el estado inicial de la tercera estación de la planta Festo : [https://www.youtube.com/watch?v=U\\_fvNbaVsVw](https://www.youtube.com/watch?v=U_fvNbaVsVw). ¿Cuál de los siguientes networks cumple las condiciones iniciales?

Tiempo de respuesta: 120  
 Respuesta correcta: 0  
 Dificultad: 0.0  
 Adivinanza: 0.5

Imagen:

Keywords: Propuestos:

Creado por: Reymar Andrea Vargas Iniesta  
 Asignatura: Automatización Industrial Avanzada

**Respuestas**  
 El uso de la marca SM0.1 implica que sólo se pondrá a 1 en el primer ciclo. Si se utilizan bobinas de asignación simple, la acción se realizará sólo una vez. El elemento vertical no se retrae. Por lo que la opción "a" es la correcta. La opción "b" es la correcta porque sólo es necesario que la línea se cumpla una sólo vez y es al principio del ciclo, después no hace falta.

Figura 7.30 Preguntas con enlace a Youtube

Cuando el alumno se dirija al enlace aparecerá el video de las condiciones iniciales de la tercera estación. Luego el alumno debe responder a la pregunta que se le planteó.

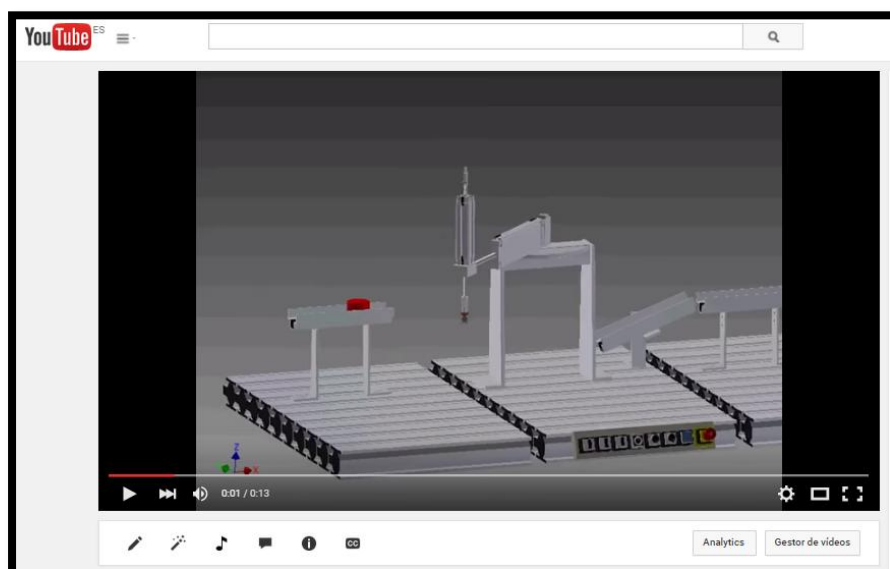


Figura 7.31 Vídeo del estado inicial de la estación

## Capítulo 8. Conclusiones

Para la implementación del método de enseñanza de la ingeniería, la primera idea era crear el software educativo haciendo uso del lenguaje de programación Python. No obstante, debido al extenso contenido sintáctico del programa, necesario para una comprensión eficaz de tal lenguaje, resultó imposible la consecución de tal objetivo.

Las dificultades con las que nos encontramos a la hora de programar en Python, nos hace pensar que, como sugerencia de mejora, quizás sería adecuado una mayor profundización en este lenguaje de programación en la asignatura correspondiente a este temario en el plan de estudios del Grado.

Gracias a la interacción en el software educativo SIENA, podemos concluir que es una herramienta informática con gran valor didáctico. El usuario puede definir los objetivos a alcanzar por el estudiante, además permite a los alumnos el estudio de una forma autónoma logrando profundizar sus conocimientos en un tema determinado.

Como parte del temario del método de aprendizaje, en este proyecto se dio a conocer un entorno de programación para PLC's muy conocido en el mundo de la automatización y desconocido para muchos estudiantes del grado de Ingeniería Electrónica Industrial y Automática. El Software CoDeSys es un programa que cumple estrictamente con el estándar Internacional IEC 61131-3, en él se pueden efectuar códigos de automatización en los cuatro lenguajes homologados, su entorno de trabajo es muy intuitivo y fácil de utilizar. El objetivo de este punto era ampliar los conocimientos en los entornos de programación de PLC's, además de animar a los futuros alumnos a que investiguen sobre la gran variedad de programas relacionados con la automatización industrial que existen en el mercado a parte de los estudiados en la carrera.

Dada la amplia utilización del Software CoDeSys, creemos que podría ser conveniente el estudio del mismo como complemento al programa de la asignatura involucrada en el desarrollo de este proyecto.

Mediante el estudio y análisis de la Estación de acarreo de piezas, se logró uno de los objetivos marcados, la implementación del código de automatización de dicha estación. Durante este proceso se profundizó los conocimientos en la programación de autómatas

programables, adquiridos a lo largo de la carrera en las asignaturas de automatización industrial.

Se realizó exitosamente la Simulación de la Estación de acarreo de piezas de la planta Festo. Para ello se hizo un estudio entre varios programas de modelado en 3D y teniendo en cuenta unos requisitos específico, se optó por la utilización del software Autodesk Inventor. Gracias a esta etapa se aprendió a trabajar con programas de modelación CAD 3D y diseño paramétrico. Aspecto muy importante, actualmente el uso de estos programas hace parte del mercado industrial, estos permiten modelar, visualizar y simular procesos reales. Para los alumnos recién graduados es un punto a favor, ya que las empresas valorarán su conocimiento y destreza en estos programas.

## Conclusion

For the implementation of the engineering teaching method, the first idea was to create the educational software using the programming language 'Python'. However, due to the program's large syntactic content, necessary for an efficient understanding of the language, it became impossible for us to accomplish such goal.

The difficulties we faced when we had to program in Python, brought us to the conclusion that, as an improvement proposal, maybe a more extensive learning of this programming language in the corresponding subject of the degree's syllabus would be convenient.

The interaction on the educational software 'SIENA' makes it, as we see it, a software tool of great educational value. The users can define the goals they want their students to achieve, and it also allows the students to study in a more autonomous way, broadening their knowledge on a given topic.

As part of the teaching method's syllabus, in this project we presented a programming environment for PLC's that is very well known in the automation world, but not so familiar to many students of the Industrial Electronic Engineering and Automation degree. CoDeSys is a program that strictly stands by the IEC 61131-3 standards, and in which we can develop automation codes in the four homologated languages, with a working environment that is both intuitive and easy to use. The goal of this particular task was to broaden the knowledge on PLC'S programming environments, as well as encouraging future students to do some research about the great variety of programs related to industrial automation that exist in the market, other than those we have studied throughout the course of the career.

Given the wide use of CoDeSys, we think it might be convenient to study how it works as a complement to the syllabus of the subject directly involved in the development of this project.

Through the study and analysis of the piece carriage station, we achieved one of the goals set, the implementation of the automation code of such station. During this process we deepen our knowledge on the programming of programmable automatons, knowledge acquired throughout the course of the career in the subjects related to industrial automation.

We successfully brought into being the simulation of the piece carriage station of the Festo plant. To do this we carried out a study between several 3D modeling programs and, with some specific requirements in mind, we opted to use the Autodesk Inventor software. This particular stage of the project helped us learn how to work with the CAD 3D modeling programs and also taught us about parametrical design. This is a very important feat, since the use of these programs is a big part of the current industrial market, allowing to model, visualize and simulate real processes. For students that have just graduated it is definitely a plus, since the different companies they might be interested in working for will certainly appraise their knowledge and skills concerning the use of these programs.



## Referencias bibliográficas

- [1] Principios de Chickering y Gamson:  
<http://repositorio.bib.upct.es/dspace/bitstream/10317/1113/1/nmd.pdf>
- [2] Simulación. Disponible en la URL:  
<http://www.etnassoft.com/biblioteca/simulacion-conceptos-y-programacion/>
- [3] Aplicación de la simulación. Disponible en la URL:  
<http://repositorio.utn.edu.ec/bitstream/123456789/621/1/Tesis.pdf>
- [4] IEC 61131-3: Recurso de programación estándar. Disponible en la URL:  
[www.plcopen.org](http://www.plcopen.org)
- [5] Beckhoff Information System, Lista de instrucciones (IL). Disponible en la URL:  
[http://infosys.beckhoff.com/espanol.php?content=../content/1034/tcpliccontrol/html/tcplicctrl\\_languages%20il.htm&id=](http://infosys.beckhoff.com/espanol.php?content=../content/1034/tcpliccontrol/html/tcplicctrl_languages%20il.htm&id=)
- [6] Jhon k. Heinz, Michael Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*, Segunda Edición. Springer. 2010.
- [7] Página oficial de CoDeSys. Disponible URL: <http://www.codesys.com/the-system.html>
- [8] Opción de Ayuda del Programa CoDeSys Versión 2.3
- [9] Página oficial VariCAD. Software de CAD 3D / 2D para la ingeniería mecánica. Disponible en la URL: <https://www.varicad.com/en/home/>
- [10] Sistemas, modelos y simulación. Universidad de Buenos Aires. Disponible en la URL: <http://materias.fi.uba.ar/7526/docs/teoria.pdf>
- [11] Ayuda Inventor Autodesk. Disponible en la URL:  
<http://help.autodesk.com/view/INVENTOR/2015/ESP/?guid=GUID-EBD2C784-0E71-452A-B5F2-22738D22D45A>
- [12] Programa interactivo. Disponible en la URL:  
<http://raulavila123.blogspot.com.es/2011/09/programas-secuenciales-interactivos-y.html>

- [13] Python. Disponible en la URL:  
<http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>
- [14] Pygame. Disponible en la URL:  
<http://www.fing.edu.uy/tecnoinf/mvd/cursos/vj2d/material/vj2d-clase05b-IntroPythonPygame.pdf>
- [15] Primer video juego. Disponible en la URL:  
<http://jonathanmelgoza.com/blog/primer-juego-con-pygame-en-python-tutorial/>
- [16] SIENA. Disponible en la URL: <http://sienasocial.ull.es/>
- [17] CompendiumLD. Disponible en la URL: <http://es.slideshare.net/sirear/nodos-e-iconos-de-compendiumld>
- Guion de Practicas de la tercera estación de la asignatura Automatización Industrial Avanzada.
- Shannon R, Johannes JD. *Systems simulation: the art and science*.1976.
- Luis R. Izquierdo, José M. Galán, José I. Santos y Ricardo del Olmo  
*Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas*.2008.
- Autómatas Programables: Introducción al Estándar IEC-61131. Disponible en la URL:  
<http://isa.uniovi.es/docencia/IngdeAutom/transparencias/Pres%20IEC%2061131.pdf>
- Programación PLC, Universidad Nacional Abierta y a Distancia. Disponible en la URL:  
[http://datateca.unad.edu.co/contenidos/2150512/ContenidoLinea/leccin\\_1632\\_diagrama\\_de\\_bloques\\_funcionales\\_function\\_block\\_diagram\\_fbd.html](http://datateca.unad.edu.co/contenidos/2150512/ContenidoLinea/leccin_1632_diagrama_de_bloques_funcionales_function_block_diagram_fbd.html)
- Solidworks. Software de CAD 3D para diseño mecánico. Disponible en la URL  
<http://www.solidworks.es/sw/products/3d-cad/capabilities.htm>
- CATIA software para diseño y desarrollo de productos 3D y PLM. Disponible en la URL: <http://www.3ds.com/products-services/catia>
- Características del software Inventor. Disponible en la URL:  
<http://www.asidek.es/industria-y-fabricacion/autodesk-inventor/>