

SECRETARÍA DE EDUCACIÓN PÚBLICA

**DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ**

REPORTE FINAL DE RESIDENCIA PROFESIONAL

**“SISTEMA PARA MONITOREO Y CONTROL DE UN ANIMATRONIC VÍA WIFI,
ETAPA DE SOFTWARE”**

INTEGRANTE (S):

No. Control	NOMBRE	Semestre
13270897	YOEL ISAAC VELASCO TOVILLA	11

PERIODO

AGOSTO-DICIEMBRE 2018

CARRERA:

INGENIERIA EN ELECTRONICA

ASESOR INTERNO:

MC. ALDO ESTEBAN AGUILAR CASTILLEJOS

ENERO 2019

Contenido	
CAPITULO I	3
1.1 Introducción	3
1.2 Problemática.....	4
1.3 Justificación	4
1.4 Objetivos	4
1.5 Alcance y Limitaciones	4
CAPITULO II	6
2.1 Caracterización Del Área En Que Se Participó.....	6
2.1.1 Antecedentes de la empresa	6
2.1.2 Organigrama de la Empresa	7
2.1.3 Misión, Visión y Valores	7
2.1.4 Descripción Del Área Donde Se Realizó El Proyecto	8
CAPITULO III	9
3.1 Fundamento Teórico	9
3.2 Conceptos básicos	15
CAPITULO IV	28
4.1 Procedimientos Y Descripción De Las Actividades Realizadas.....	28
4.2 Implementación.....	30
4.2.1 Programación del PLC.....	33
4.2.2 Programación de la página de usuario en HTML.....	39
CAPITULO V	45
REFERENCIAS	47

CAPITULO I

1.1 Introducción

El proyecto “Sistema Para Monitoreo Y Control De Un Animatronic Vía Wifi, Etapa De Software”. Fue desarrollado en el Instituto Tecnológico De Tuxtla Gutiérrez. Donde se buscó poder actualizar el Animatronic, de manera que fuese posible poder obtener datos del mismo de manera remota por medio del puerto ethernet. Todo esto con el fin de poder monitorear y manejar el mismo a distancia para presentaciones en las que el ponente no pueda estar presente y recabar información y datos para su mejora.

En primera instancia se planteó la problemática que había en cuanto al manejo del animatronic, ya que para su exposición era necesario que el operador estuviese presente en la sala y de manera manual accionara los interruptores que permitían darle movimiento al animatronic. Desafortunadamente en frecuentes ocasiones era necesaria un control del mismo vía web por medio de una página elaborada por el usuario que permitiera su operación remota para presentaciones a distancia de igual manera con el fin de actualizar el sistema que se tenía actualmente.

En el presente documento se desglosa cada uno de los conceptos que fueron necesarios para tal actualización, de igual manera el proceso que se llevó a cabo para el buen funcionamiento del mismo.

1.2 Problemática

En nuestra institución contamos con muchos proyectos desarrollados por alumnos y docentes que han tenido el privilegio de ir a eventos nacionales de innovación. Estos proyectos son expuestos en diferentes foros con ponentes presenciales en los que se motivan a los jóvenes estudiantes a innovar e invertir su tiempo y habilidades en el desarrollo de nuevas tecnologías que agilicen un proceso implementando sistemas mecatrónicas. Uno de estos proyectos que se exponen con frecuencia es un "Jaguar Animatronic". Este prototipo ha sido desarrollado por alumnos del Instituto Tecnológico De Tuxtla Gutiérrez con la asesoría de docentes académicos de la misma. Desafortunadamente cuenta con ciertas limitaciones que muchas veces han impedido la exposición del mismo en distintos foros debido a que no es posible poder tener un monitoreo inalámbrico. Esto es necesario implementarlo ya que los ponentes muchas veces no pueden hacer una exposición presencial por ciertas cuestiones. Por esta razón es que es necesario implementar una estructura programable que le permita al usuario poder manipular, monitorear y recabar datos del mismo en tiempo real con el fin de poder tener la facilidad de ser expuesto de manera remota sin la presencia del operador.

1.3 Justificación

En la actualidad contamos con herramientas suficientes para tener el control casi total de cualquier componente electrónico a distancia. Gracias a estas herramientas podemos ahorrar tiempo y energía e invertirla en algo mas productivo. Viendo la necesidad de poder tener el control en la palma de la mano, se nos es necesario actualizar nuestros equipos electrónicos para poder estar a la vanguardia de las nuevas tecnologías y poder brindar un servicio de calidad.

1.4 Objetivos

Diseñar, programar e implementar la comunicación Wifi entre un móvil y el controlador lógico programable para la adquisición de datos.

1.5 Alcance y Limitaciones

Este mismo sistema puede ser implementado en diversos proyectos y con una gran facilidad, ya que no es tan necesario de un conocimiento demasiado amplio en programación para hacer este tipo de comunicaciones. El sistema que se diseñó no solamente nos permite controlar al Animatronic ya que se tiene el control completo del pc por medio de Escritorio remoto de Chrome podemos acceder a la programación tanto del plc como de la página que lo controla. Hacer modificaciones al mismo y grabarla en el plc.

En cuanto a las limitaciones, es necesario tener siempre un equipo de cómputo enlazado al proyecto y a internet el cual funciona como servidor, este nos permite tener comunicación entre el proyecto y nuestro controlador remoto.

CAPITULO II

2.1 Caracterización Del Área En Que Se Participó

2.1.1 Antecedentes de la empresa

El Instituto Tecnológico de Tuxtla Gutiérrez brinda Servicios Educativos de Calidad Certificada, con la misión de formar de manera integral a profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al ambiente y apego a los valores éticos.

Somos una institución de excelencia en la educación superior tecnológica del sureste, comprometidos con el desarrollo sustentable de la región, el cual es uno de los ejes transversales de nuestra oferta educativa, tanto a nivel licenciatura como posgrado.

Trabajamos con Sistemas de Gestión Ambiental Certificados, en donde nuestros estudiantes aplican sus conocimientos en función del cuidado del ambiente: aire, agua, suelo, flora, fauna y seres humanos.

La sociedad del nuevo milenio se enfrenta a retos y cambios continuos, choques generacionales en el uso de modernas tecnologías, comunicándonos en tiempo real a cualquier parte del mundo con video y voz, estamos ante un sinnúmero de avances tecnológicos que son punto de partida para modelar el desarrollo de la sociedad.

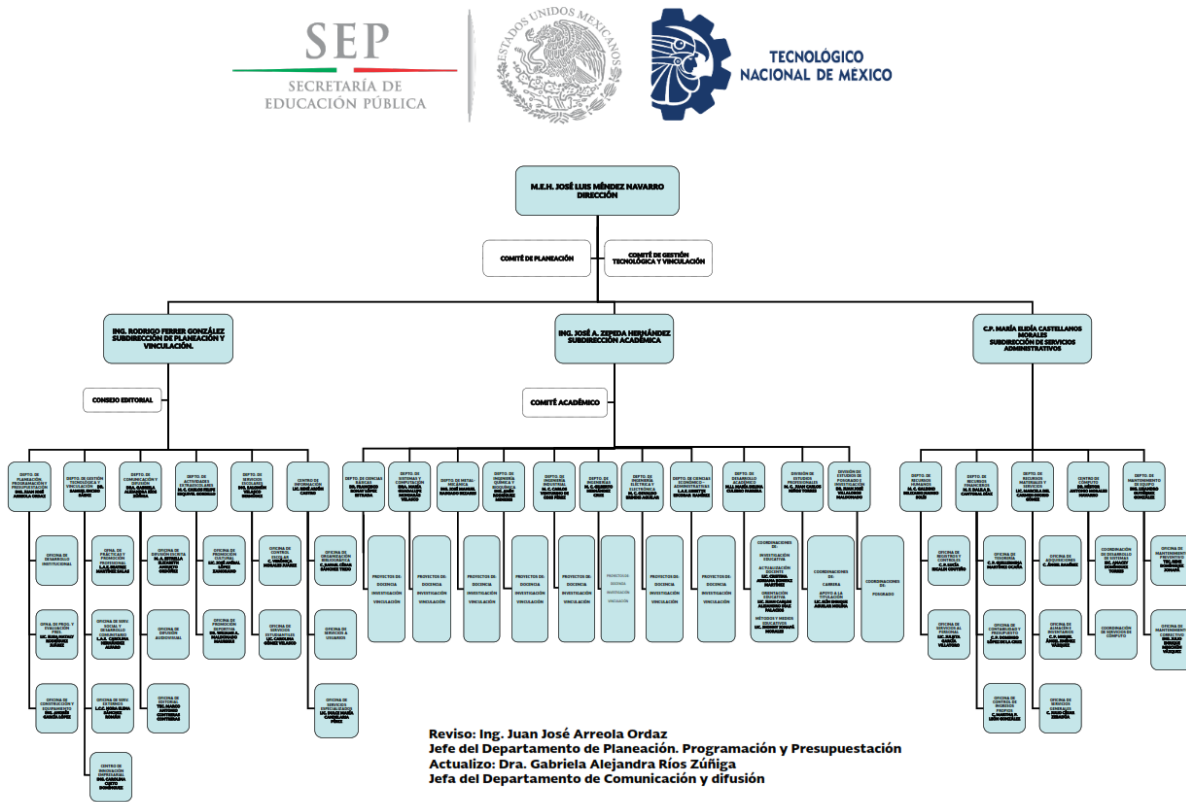
La búsqueda de un desarrollo que permita satisfacer las necesidades básicas y las aspiraciones de bienestar de la población, sin comprometer la posibilidad de las generaciones futuras de atender a sus propias necesidades y aspiraciones, se ha integrado a la ideología de nuestra institución para enriquecer las decisiones con visión de futuro.

Nosotros somos congruentes con la misión que hemos diseñado es por ellos que nos hemos ocupado por Educar con Responsabilidad Ambiental,

comprometiéndonos con un movimiento interinstitucional que genere la participación de todos los actores del instituto para con el desarrollo sustentable de la entidad.

Entre las principales líneas estratégicas de Educar con Responsabilidad Ambiental se encuentran los Sistemas de Gestión de Calidad, Sistema de Gestión Ambiental, Programas Académicos Acreditados por COPAES y por CONACYT; se trata de transformar conciencias, evolucionar culturas, sensibilizar a la población y mostrar responsabilidad ante las acciones que realizamos en pro de la conservación de la biodiversidad del planeta.

2.1.2 Organigrama de la Empresa



2.1.3 Misión, Visión y Valores

Valores

El Ser Humano

El Espíritu de Servicio

El Liderazgo

El Trabajo en Equipo

La Calidad

El Alto Desempeño

Respeto al Medio Ambiente

Misión

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

Visión

Ser una Institución de Excelencia en la Educación Superior Tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

2.1.4 Descripción Del Área Donde Se Realizó El Proyecto

Este proyecto fue realizado en las instalaciones del Instituto Tecnológico De Tuxtla Gutiérrez. Este instituto cuenta con laboratorios suficientemente equipados para pruebas, creación y manipulación de elementos electrónicos, informáticos y mecánicos. Un excelente ambiente de trabajo con académicos muy bien preparados a los cuales podemos consultar y resolver cualquier duda que se nos presente a lo largo del desarrollo del proyecto.

CAPITULO III

3.1 Fundamento Teórico

Controlador lógico programable (PLC)

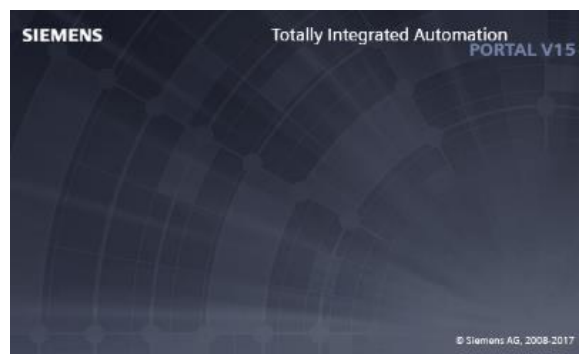


Un controlador lógico programable, más conocido por sus siglas en inglés PLC (Programmable Logic Controller) o por autómatas programables, es una computadora utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas.

Los PLC son utilizados en muchas industrias y máquinas. A diferencia de las computadoras de propósito general, el PLC está diseñado para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto. Los programas para el control de funcionamiento de la máquina se suelen almacenar en baterías, copia de seguridad o en memorias no volátiles. Un PLC es un ejemplo de un sistema de tiempo real «duro contra el muro», donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, de lo contrario no producirá el resultado deseado.

TIA Portal

TIA Portal es el innovador sistema de ingeniería que permite configurar de forma intuitiva y eficiente todos los procesos de planificación y producción. Convince por

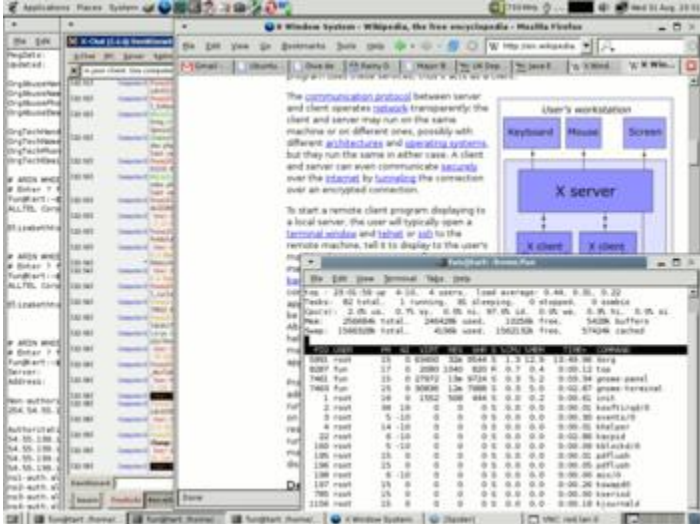


su funcionalidad probada y por ofrecer un entorno de ingeniería unificado para todas las tareas de control, visualización y accionamiento.

El TIA Portal incorpora las últimas versiones de Software de Ingeniería SIMATIC STEP 7, WinCC y Startdrive para la planificación, programación y diagnóstico de todos los controladores SIMATIC, pantallas de visualización y accionamientos SINAMICS de última generación.

Escritorio remoto.

Un escritorio remoto es una tecnología que permite a un usuario trabajar en una computadora a través de su escritorio gráfico desde otro dispositivo terminal ubicado en otro lugar.



La tecnología de escritorio remoto permite la centralización de aquellas aplicaciones que generalmente se ejecutan en entorno de usuario (por ejemplo, procesador de textos o navegador). De esta manera, dicho entorno de usuario se transforma en meros terminales de entrada/salida.

Los eventos de pulsación de teclas y movimientos de ratón se transmiten a un servidor central donde la aplicación los procesa como si se tratase de eventos locales. La imagen en pantalla de dicha aplicación es retornada al terminal cliente cada cierto tiempo.

Telecomunicación



Una telecomunicación es toda transmisión y recepción de señales de cualquier naturaleza, típicamente electromagnéticas, que contengan signos, sonidos, imágenes o, en definitiva, cualquier tipo de información que se desee comunicar a cierta distancia.

Por metonimia, también se denomina telecomunicación (o telecomunicaciones, indistintamente) a la disciplina que estudia, diseña, desarrolla y explota aquellos sistemas que permiten dichas comunicaciones; de forma análoga, la ingeniería de telecomunicaciones resuelve los problemas técnicos asociados a esta disciplina.

Las telecomunicaciones son una infraestructura básica del contexto actual. La capacidad de poder comunicar cualquier orden militar o política de forma casi instantánea ha sido radical en muchos acontecimientos históricos de la Edad Contemporánea —el primer sistema de telecomunicaciones moderno aparece durante la Revolución Francesa—. Pero además, la telecomunicación constituye hoy en día un factor social y económico de gran relevancia. Así, estas tecnologías adquieren una importancia como su utilidad en conceptos de la globalización o la sociedad de la información y del conocimiento; que se complementa con la importancia de las mismas en cualquier tipo de actividad mercantil, financiera, bursátil o empresarial. Los medios de comunicación de masas también se valen de las telecomunicaciones para compartir contenidos al público, de gran importancia a la hora de entender el concepto de sociedad de masas.

La telecomunicación incluye muchas tecnologías como la radio, televisión, teléfono y telefonía móvil, comunicaciones de datos, redes informáticas, Internet, radionavegación o GPS o telemetría. Gran parte de estas tecnologías, que nacieron para satisfacer necesidades militares o científicas, ha convergido en otras enfocadas a un consumo no especializado llamadas tecnologías de la información y la comunicación, de gran importancia en la vida diaria de las personas, las empresas o las instituciones estatales y políticas. Es por este contexto que la

tendencia actual es la comunión de la telecomunicación con otras disciplinas como la informática, la electrónica o la telemática para diseñar y ofrecer estos productos y servicios, lo suficientemente complejos y multidisciplinarios como para que la frontera entre la aportación de dichas disciplinas no sea percibida por las personas —a pesar de que un informático, un eléctrico y un telecomunicador tengan distintos ámbitos disciplinarios—.

Animatrónica

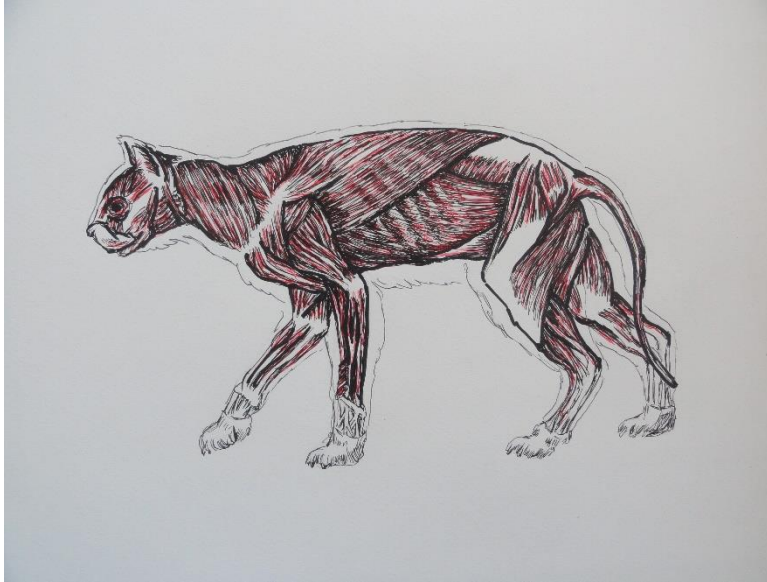
La animatrónica es la técnica que, mediante el uso de mecanismos robóticos o electrónicos, simula el aspecto y comportamiento de los seres vivos empleando marionetas u otros muñecos mecánicos. Se caracterizan por tener un aspecto físico antropomórfico. Son creados para ser programados y controlados remotamente, reproducir sonido y recrear movimientos ya sean sencillos o de gran complejidad. La sofisticación de estos robots depende del uso o servicio que vayan a cumplir, ya que, por ejemplo, se puede recrear únicamente el lomo de un oso o crear todo el animal completo.



Anatomía del jaguar

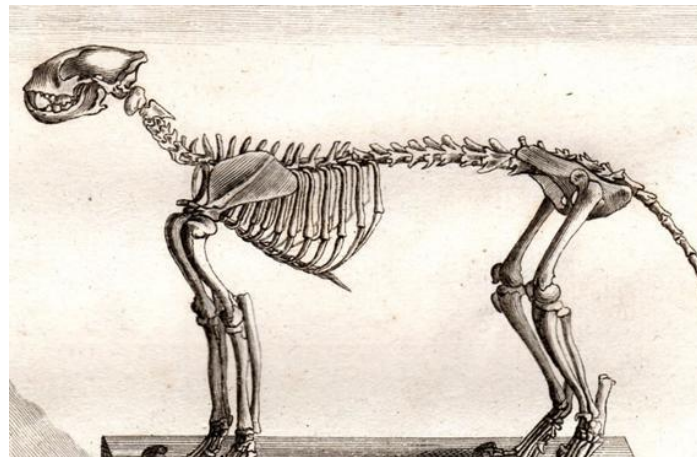
El jaguar es el felino más grande de las Américas y el tercer más grande del mundo, por detrás del tigre y del león. Mantiene cierta semejanza física con el leopardo (*Panthera pardus*) pero es una especie única, con marcadas características. ¿Quieres conocerlo?

Lo primero que debes de saber es que el carnívoro más grande de América mide de 1.2 a 1.95 metros de longitud hasta la base de la cola, es decir, casi 2 metros de largo. Eso es seguramente mayor que tu propia estatura. Sin embargo, su cola es relativamente corta: entre 45 y 75 centímetros de longitud. Su altura hasta los



hombros es de 63-76 centímetros. Existe dimorfismo sexual, es decir, diferencias físicas entre los sexos, puesto que la hembra es un 10-20 por ciento más pequeña que el macho.

El pelaje del jaguar es quizá lo más llamativo de todo su cuerpo. Es dueño de un corto pelaje que va de marrón dorado a amarillo pálido o rojizo. Este fondo claro contrasta con las rosetas oscuras, manchas redondas cuyo centro es de un amarillo



o marrón más intenso y presenta puntos oscuros más pequeños. Las rosetas difieren de las del leopardo, ya que éste las posee sin puntos pequeños. El cuello, la región ventral y las zonas internas de las extremidades son blanquecinos mientras que la cola se torna blanca hacia el final y negra en la punta.

Algunos jaguares nacen con un pelaje oscuro o completamente negro. Se debe a una mayor producción del pigmento melanina y no a una subespecie diferente. Aún más raros son los jaguares blancos, conocidos como panteras blancas. Ahora bien, es momento de reconocer la forma de su cuerpo: Joven jaguar de color negro, generalmente conocido como pantera negra.

Cabeza

Es grande, robusta y más redondeada que la del leopardo. Posee poderosas mandíbulas cuadradas y mejillas sobresalientes; su mordida es la más fuerte de todos los felinos y le permite romper con sorprendente facilidad los caparazones de las tortugas y traspasar la cubierta protectora de los armadillos.

Los ojos son redondos, con pupilas en forma de círculos pequeños y el iris rojo o rojo amarillento. Dado que los ojos están posicionados al frente del rostro posee una visión binocular útil para seguir la pista de sus presas. Su vista es unas 6 veces más aguda que la de un ser humano.

Dentro de la boca emergen dos caninos superiores largos y sumamente afilados cuya especialidad es la perforación del hueso y de materiales muy duros. Sus premolares son también muy afilados para cortar la carne.

Torso

Gracias a su constitución corta, compacta y musculosa, puede escalar y nadar con mucha facilidad. En general, su cuerpo está construido para hacer uso de la fuerza y poder en vez de la velocidad.

El tamaño del cuerpo de los jaguares puede variar en función de su hábitat y la región geográfica donde se encuentra. Por ejemplo, los individuos del Pantanal de Brasil suelen ser más grandes que los de otras partes del mundo, y aquellos que prosperan en bosques de vegetación muy densa son más pequeños que quienes viven en hábitats un poco más abiertos. En adición, el tamaño de la presa típica también incide en el tamaño del cuerpo de los jaguares.

Extremidades

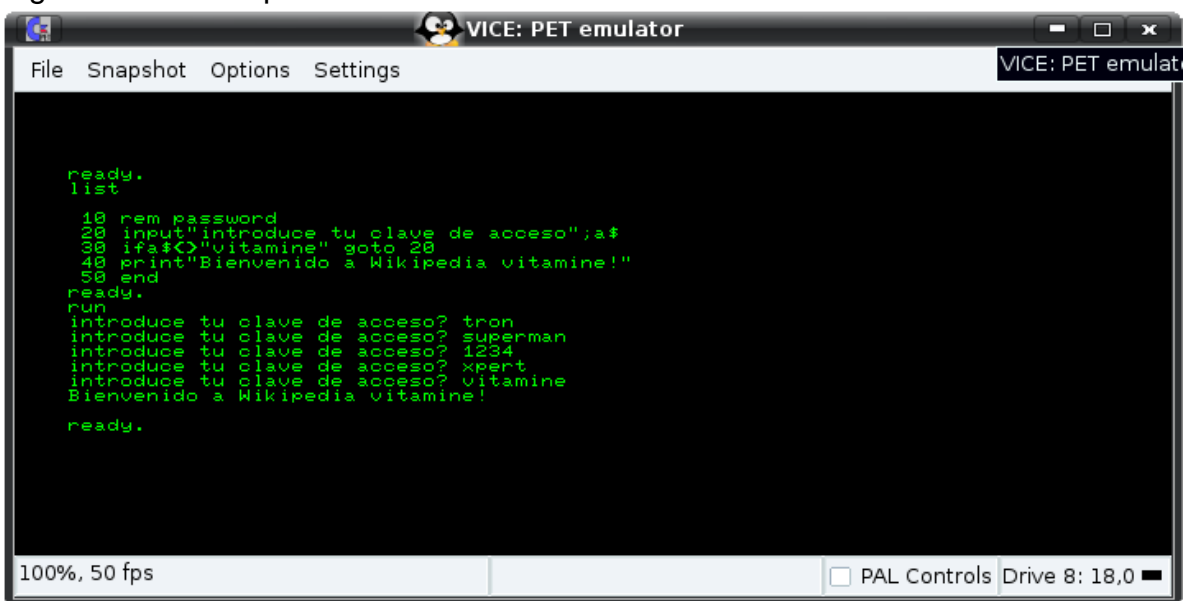
Tanto las piernas como la cola son cortas en comparación con las de otras especies de felinos. Esto no impide que los músculos le provean fuerza al correr o saltar mientras que las patas largas confieren estabilidad. Sus patas traseras son las más cortas de todos los felinos del género Panthera.

Sus garras retráctiles, su cuerpo corto y sus hombros fuertes le permiten derribar presas tan grandes como tapires. ¡Es todo un depredador!

3.2 Conceptos básicos

Lenguaje de programación

Un lenguaje de programación es un lenguaje formal que especifica una serie de instrucciones para que una computadora produzca diversas clases de datos. Los lenguajes de programación pueden usarse para crear programas que pongan en práctica algoritmos específicos los cuales controlan el comportamiento físico y lógico de una computadora.



The image shows a screenshot of the VICE: PET emulator window. The window title is "VICE: PET emulator" and it has a menu bar with "File", "Snapshot", "Options", and "Settings". The main area is a black terminal window with green text. The text shows a BASIC program being executed. The program starts with "ready." and "list", then lists several lines of code: "10 rem password", "20 input \"introduce tu clave de acceso\";a\$", "30 if a\$<> \"vitamine\" goto 20", "40 print \"Bienvenido a Wikipedia vitamine!\"", and "50 end". The program then runs, showing "ready." and "run". The output shows the program asking for a password: "introduce tu clave de acceso? tron", "introduce tu clave de acceso? superman", "introduce tu clave de acceso? 1234", "introduce tu clave de acceso? xpert", and "introduce tu clave de acceso? Vitamine". The program then prints "Bienvenido a Wikipedia vitamine!". The terminal window has a status bar at the bottom showing "100%, 50 fps" and "PAL Controls Drive 8: 18,0".

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

El desarrollo lógico del programa para resolver un problema en particular.

Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).

Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.

Prueba y depuración del programa.

Desarrollo de la documentación.

```
1 // class declaration
2 public class ProgrammingExample {
3
4     // method declaration
5     public void sayHello() {
6
7         // method output
8         System.out.println("Hello World!");
9     }
10 }
```

Existe un error común que trata por sinónimos los términos 'lenguaje de programación' y 'lenguaje informático'. Los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como por ejemplo HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación, sino un conjunto de instrucciones que permiten estructurar el contenido de los documentos).

Permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. Una característica relevante de los lenguajes de programación es precisamente que más de un programador pueda usar un conjunto común de instrucciones que sean comprendidas entre ellos para realizar la construcción de un programa de forma colaborativa.

Elementos

Variables y vectores

Imagen tomada de Pauscal, lenguaje de programación en español creado en Argentina.

Las variables son títulos asignados a espacios en memoria para almacenar datos específicos. Son contenedores de datos y por ello se diferencian según el tipo de dato que son capaces de almacenar. En la mayoría de lenguajes de programación se requiere especificar un tipo de variable concreto para guardar un dato específico. Por ejemplo, en Java, si deseamos guardar una cadena de texto debemos especificar que la variable es del tipo String. Por otra parte, en lenguajes como PHP o JavaScript este tipo de especificación de variables no es necesario. Además, existen variables compuestas llamadas vectores. Un vector no es más que un conjunto de bytes consecutivas en memoria y del mismo tipo guardadas dentro de una variable contenedor. A continuación, un listado con los tipos de variables y vectores más comunes:

Tipo de dato Breve descripción

Char	Estas variables contienen un único carácter, es decir, una letra, un signo o un número.
Int	Contienen un número entero.
Float	Contienen un número decimal.
String	Contienen cadenas de texto, o lo que es lo mismo, es un vector con varias variables del tipo Char.
Boolean	Solo pueden contener un cero o un uno.

En el caso de variables booleanas, el cero es considerado para muchos lenguajes como el literal falso ("False"), mientras que el uno se considera verdadero ("True").

Condicionales

Las sentencias condicionales son estructuras de código que indican que, para que cierta parte del programa se ejecute, deben cumplirse ciertas premisas; por ejemplo: que dos valores sean iguales, que un valor exista, que un valor sea mayor que otro... Estos condicionantes por lo general solo se ejecutan una vez a lo largo del programa. Los condicionantes más conocidos y empleados en programación son:

If: Indica una condición para que se ejecute una parte del programa.

Else if: Siempre va precedido de un "If" e indica una condición para que se ejecute una parte del programa siempre que no cumpla la condición del if previo y sí se cumpla con la que el "else if" especifique.

Else: Siempre precedido de "If" y en ocasiones de "Else If". Indica que debe ejecutarse cuando no se cumplan las condiciones previas.

Bucles

Los bucles son parientes cercanos de los condicionantes, pero ejecutan constantemente un código mientras se cumpla una determinada condición. Los más frecuentes son:

For: Ejecuta un código mientras una variable se encuentre entre 2 determinados parámetros.

While: Ejecuta un código mientras que se cumpla la condición que solicita.

Hay que decir que a pesar de que existan distintos tipos de bucles, todos son capaces de realizar exactamente las mismas funciones. El empleo de uno u otro depende, por lo general, del gusto del programador.

Funciones

Las funciones se crearon para evitar tener que repetir constantemente fragmentos de código. Una función podría considerarse como una variable que encierra código dentro de sí. Por lo tanto cuando accedemos a dicha variable (la función) en realidad lo que estamos haciendo es ordenar al programa que ejecute un determinado código predefinido anteriormente.

Todos los lenguajes de programación tienen algunos elementos de formación primitivos para la descripción de los datos y de los procesos o transformaciones aplicadas a estos datos (tal como la suma de dos números o la selección de un elemento que forma parte de una colección). Estos elementos primitivos son definidos por reglas sintácticas y semánticas que describen su estructura y significado respectivamente.

Sintaxis

Con frecuencia se resaltan los elementos de la sintaxis con colores diferentes para facilitar su lectura. Este ejemplo está escrito en Python.

A la forma visible de un lenguaje de programación se le conoce como sintaxis. La mayoría de los lenguajes de programación son puramente textuales, es decir, utilizan secuencias de texto que incluyen palabras, números y puntuación, de manera similar a los lenguajes naturales escritos. Por otra parte, hay algunos lenguajes de programación que son más gráficos en su naturaleza, utilizando relaciones visuales entre símbolos para especificar un programa.

La sintaxis de un lenguaje de programación describe las combinaciones posibles de los símbolos que forman un programa sintácticamente correcto. El significado que se le da a una combinación de símbolos es manejado por su semántica (ya sea formal o como parte del código duro de la referencia de implementación). Dado que la mayoría de los lenguajes son textuales, este artículo trata de la sintaxis textual.

La sintaxis de los lenguajes de programación es definida generalmente utilizando una combinación de expresiones regulares (para la estructura léxica) y la Notación de Backus-Naur (para la estructura gramática). Este es un ejemplo de una gramática simple, tomada de Lisp:

expresión ::= átomo | lista

átomo ::= número | símbolo

número ::= [+]? ['0'-'9']+

símbolo ::= ['A'-'Z'] ['a'-'z']*

lista ::= '(' expresión* ')'

Con esta gramática se especifica lo siguiente:

una expresión puede ser un átomo o una lista;

un átomo puede ser un número o un símbolo;

un número es una secuencia continua de uno o más dígitos decimales, precedido opcionalmente por un signo más o un signo menos;

un símbolo es una letra seguida de cero o más caracteres (excluyendo espacios); y

una lista es un par de paréntesis que abren y cierran, con cero o más expresiones en medio.

Algunos ejemplos de secuencias bien formadas de acuerdo a esta gramática:

'12345', '()', '(a b c232 (1))'

No todos los programas sintácticamente correctos son semánticamente correctos. Muchos programas sintácticamente correctos tienen inconsistencias con las reglas del lenguaje; y pueden (dependiendo de la especificación del lenguaje y la solidez de la implementación) resultar en un error de traducción o ejecución. En algunos casos, tales programas pueden exhibir un comportamiento indefinido. Además, incluso cuando un programa está bien definido dentro de un lenguaje, todavía puede tener un significado que no es el que la persona que lo escribió estaba tratando de construir.

Usando el lenguaje natural, por ejemplo, puede no ser posible asignarle significado a una oración gramaticalmente válida o la oración puede ser falsa:

"Las ideas verdes y descoloridas duermen furiosamente" es una oración bien formada gramaticalmente pero no tiene significado comúnmente aceptado.

"Juan es un soltero casado" también está bien formada gramaticalmente pero expresa un significado que no puede ser verdadero.

El siguiente fragmento en el lenguaje C es sintácticamente correcto, pero ejecuta una operación que no está definida semánticamente (dado que p es un apuntador nulo, las operaciones p->real y p->im no tienen ningún significado):

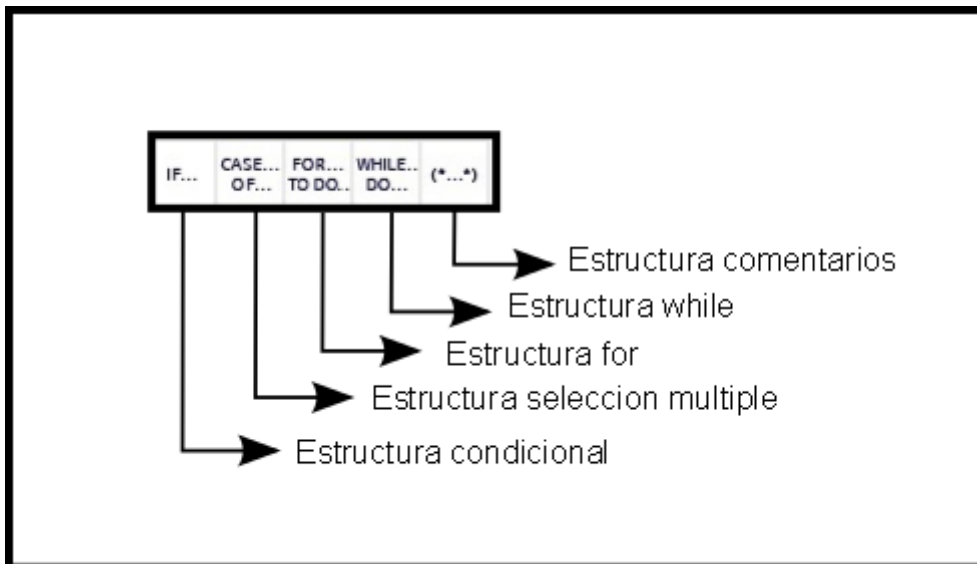
```
complex *p = NULL;
```

```
complex abs_p = sqrt (p->real * p->real + p->im * p->im);
```

Si la declaración de tipo de la primera línea fuera omitida, el programa dispararía un error de compilación, pues la variable "p" no estaría definida. Pero el programa sería sintácticamente correcto todavía, dado que las declaraciones de tipo proveen información semántica solamente.

La gramática necesaria para especificar un lenguaje de programación puede ser clasificada por su posición en la Jerarquía de Chomsky. La sintaxis de la mayoría de los lenguajes de programación puede ser especificada utilizando una gramática Tipo-2, es decir, son gramáticas libres de contexto. Algunos lenguajes, incluyendo a Perl y a Lisp, contienen construcciones que permiten la ejecución durante la fase de análisis. Los lenguajes que permiten construcciones que permiten al programador alterar el comportamiento de un analizador hacen del análisis de la sintaxis un problema sin decisión única, y generalmente oscurecen la separación entre análisis y ejecución. En contraste con el sistema de macros de Lisp y los bloques BEGIN de Perl, que pueden tener cálculos generales, las macros de C son meros reemplazos de cadenas, y no requieren ejecución de código.

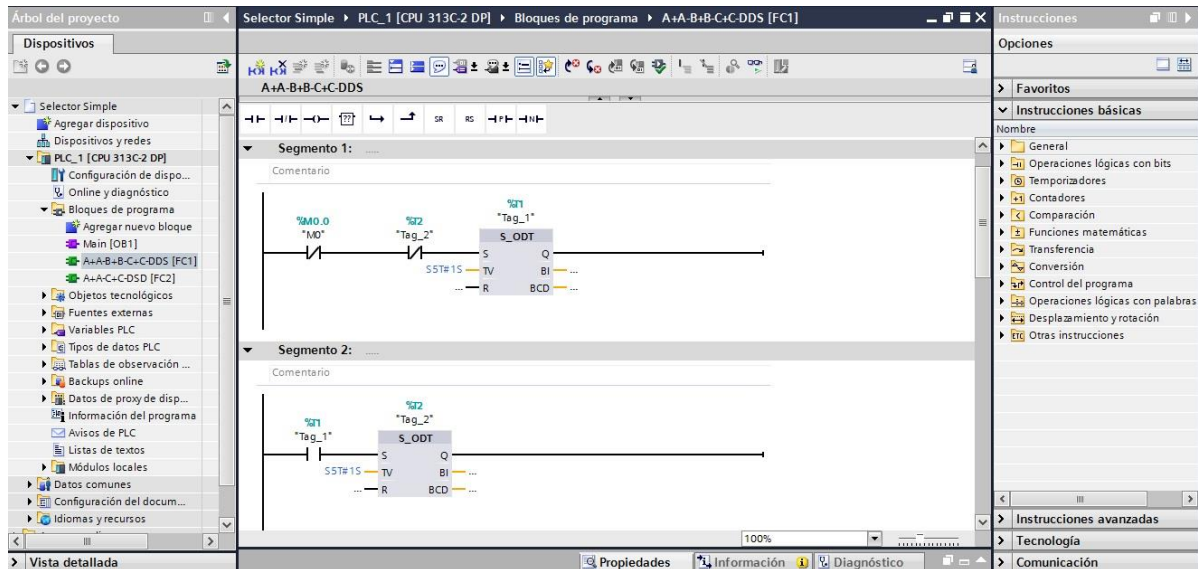
Lenguajes de programación en SIEMENS



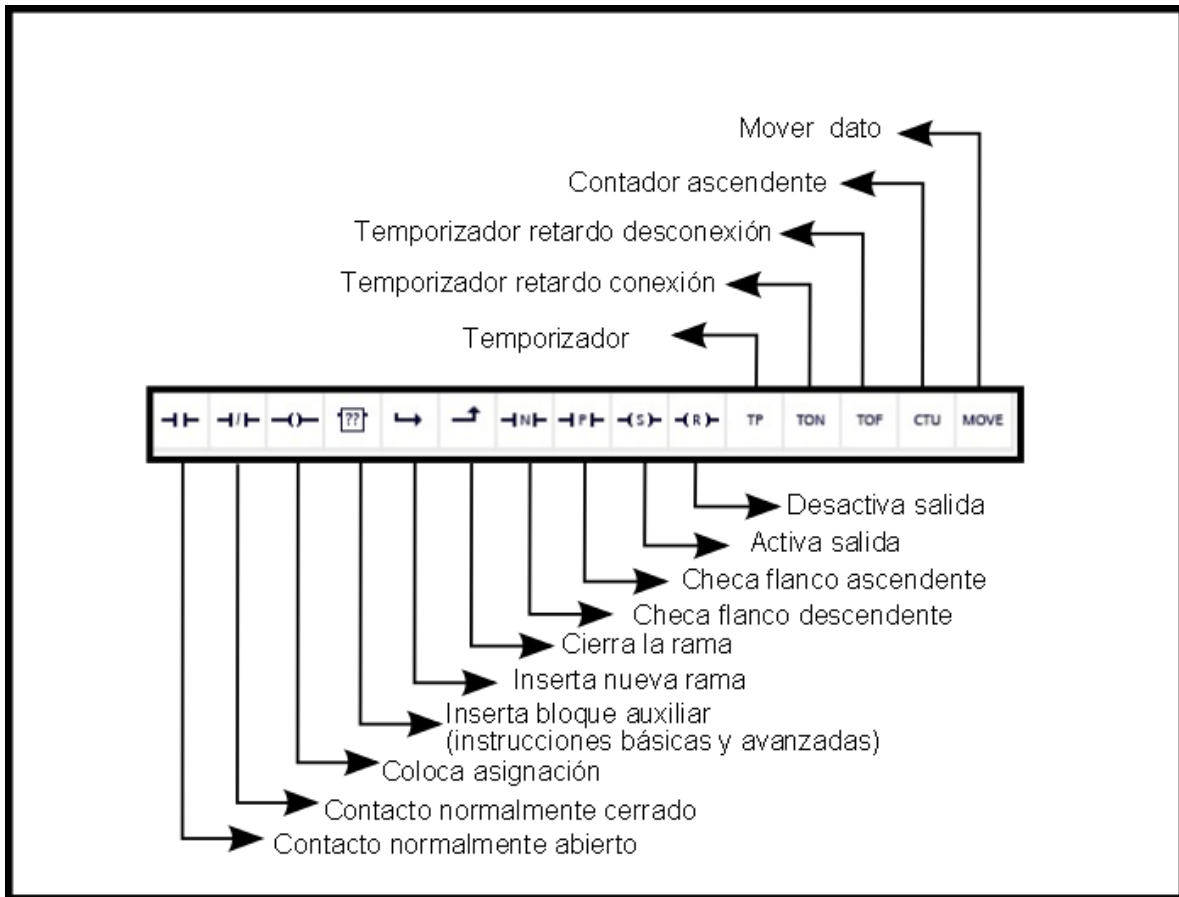
Para poder dominar cualquier controlador ya sea industrial o no, es necesario traducirle las operaciones que deseamos que realice. Esta traducción se realiza por medio de lenguajes de programación los cuales funcionan como intermediarios entre nosotros y los controladores. Dentro de los lenguajes de programación más

KOP (*Kontaktplan*)

También conocido como **diagrama de contactos o de escalera**. A diferencia del FUP, este lenguaje hace uso de lógica booleana por medio de contactos eléctricos en serie y en paralelo. Actualmente es el lenguaje más ocupado en la programación de PLC's ya que es muy fácil de entender para personas familiarizadas a diagramas eléctricos.

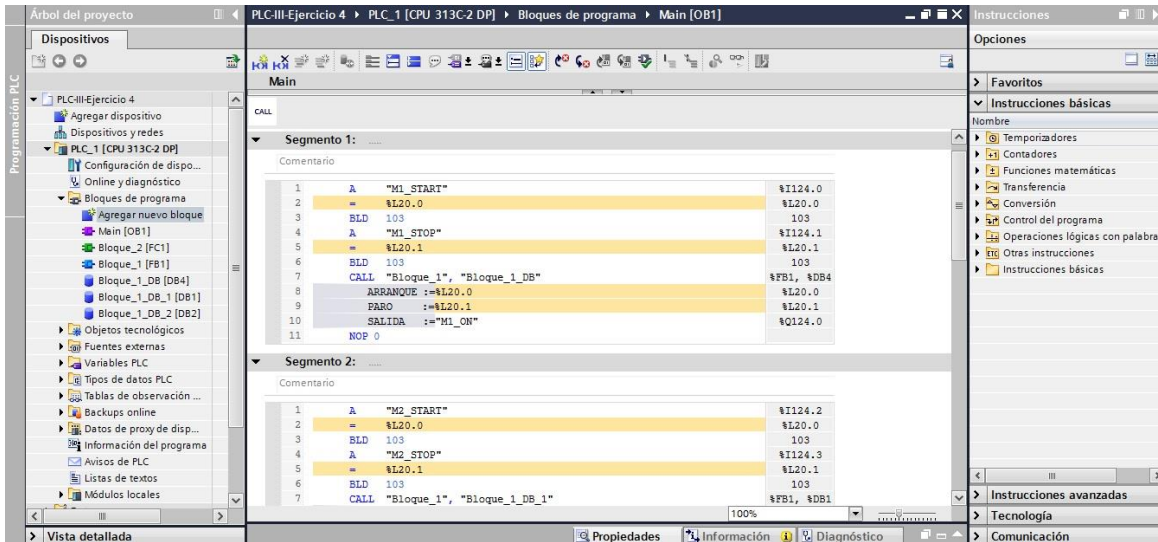


Las principales instrucciones que podemos encontrar al programar en este lenguaje son:

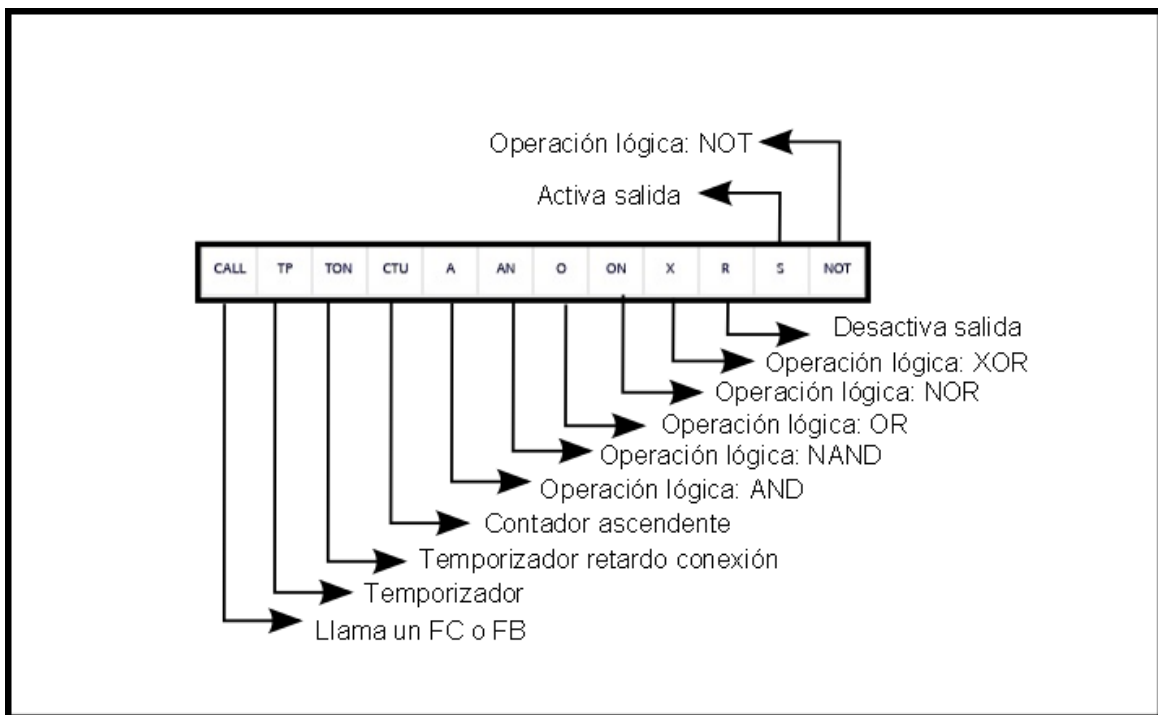


AWL (Anweisungsliste)

Denominado como **listas de instrucciones**. Este lenguaje se parece mucho a las programación utilizada en Ensamblador ya que busca dar instrucciones en un nivel muy bajo de programación para que el controlador no pierda mucho tiempo en traducir la información. La gran desventaja que existe con este lenguaje es el tamaño del código que se hace más grande mientras más complejo se hace el proceso.



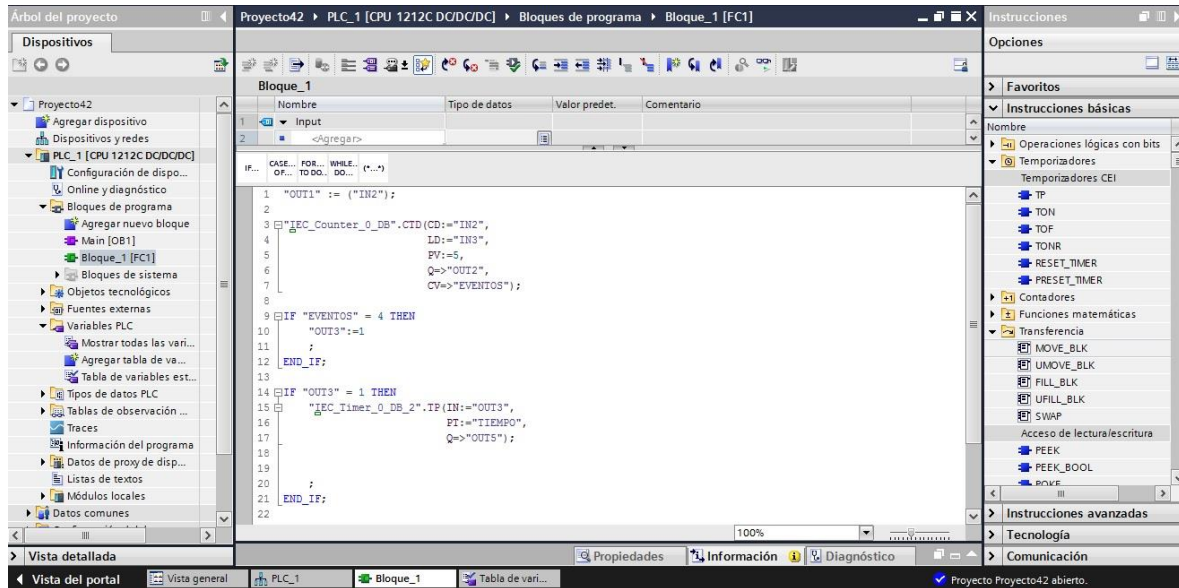
La principales instrucciones que podemos encontrar al programar en este lenguaje son:



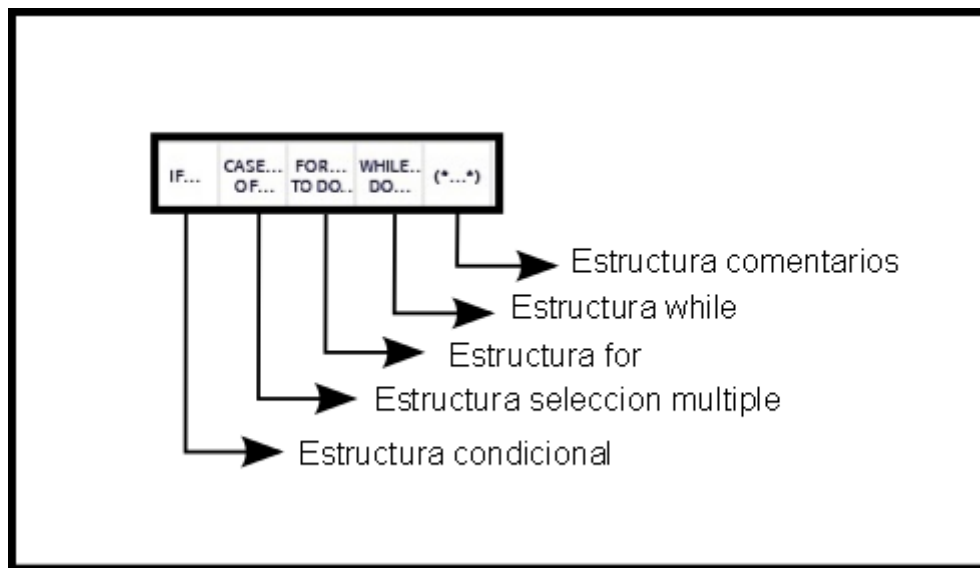
S7 – SCL (*Structured Control Language*)

A raíz de la complejidad de lectura y depuración de los programas realizados en AWL, Siemens proporciona un lenguaje extra que **se parece mucho a PASCAL, un lenguaje de alto nivel** que proporciona sentencias de bucles y condiciones. Se considera de mucha utilidad cuando se busca implementar programas con calculo

de formulas, algoritmos de optimización robustos y cuando se tenga que analizar mucha información.



Las principales instrucciones que podemos encontrar al programar en este lenguaje son:

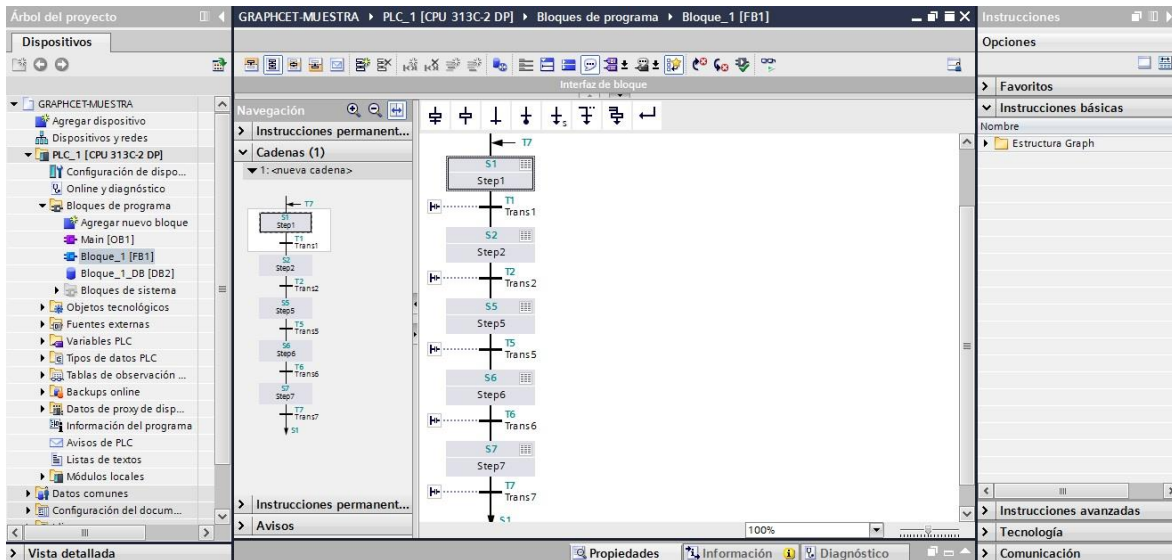


S7 – GRAPH

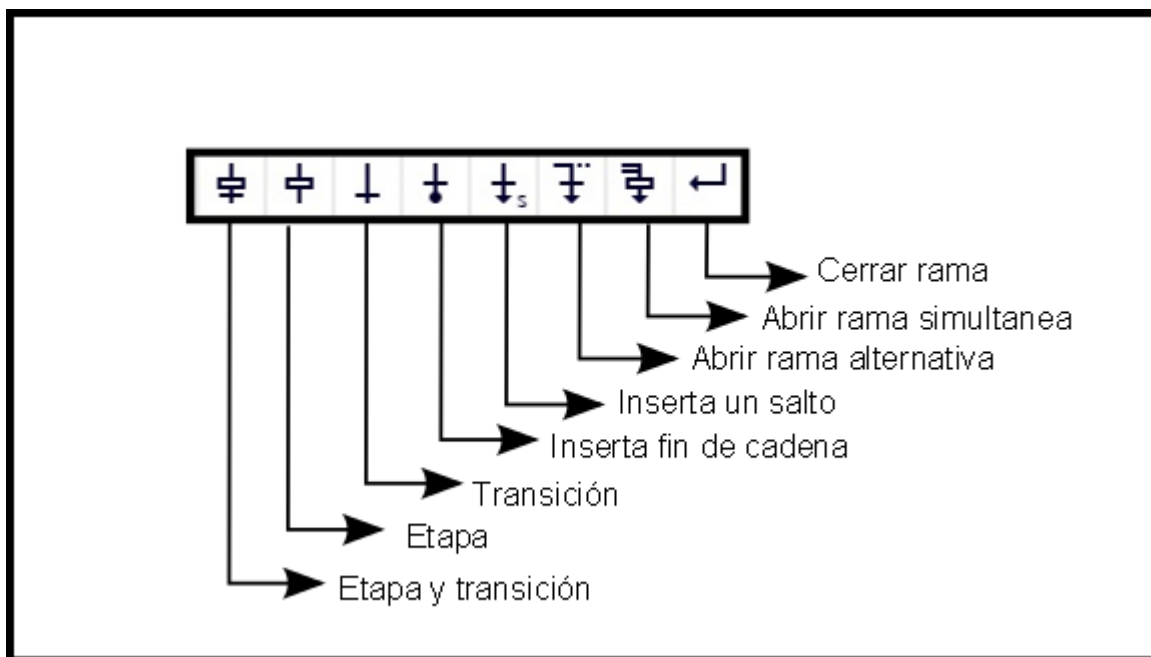
Este lenguaje funge como un **híbrido entre las lenguajes de KOP y AWL**. La función principal de este lenguaje es la creación de cadena de etapas; es decir, pequeños bloques de código que representan funciones específicas del proceso. El contenido de las etapas se define un lenguaje de programación especial (similar a AWL), en tanto que las condiciones de transición se introducen en una

representación del esquema de contactos (parte del lenguaje de programación KOP).

S7-GRAPH permite representar también procesos complejos de forma muy clara, permitiendo así una programación y una búsqueda de errores efectivas.



Las principales instrucciones que podemos encontrar al programar en este lenguaje son:



CAPITULO IV

4.1 Procedimientos Y Descripción De Las Actividades Realizadas

a) Cronograma preliminar de actividades

Actividad	Semana															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Búsqueda Bibliográfica	■	■	■													
Revisión de la programación base del animatronic			■	■	■	■										
Diseño del software o app						■	■	■	■	■						
Sincronización con el animatronic											■	■	■	■		
Última revisión														■	■	
Entrega final																■

b) Descripción detallada de las actividades

Búsquedas bibliográficas: En esta semana busque la información que me ayudara a comprender un poco más acerca de los elementos que se necesitaba para elaborar la programación, en que lenguaje y los programas que me permitirían hacer tal programación.

En primera instancia la investigación acerca del plc me permitió ahondar un poco más en el conocimiento anteriormente adquirido, me vi en la necesidad de repasar ciertos bloques que se necesitaban realizar la programación.

De igual manera en cuanto al lenguaje html que anteriormente había hecho uso en clases pasadas.

En cuanto al propósito que se quería lograr al elaborar el software fue necesario de igual manera investigar que función desempeñaba un animatronic.

Revisión de la programación base del animatronic: Las siguientes cuatro semanas se ocupó para poder familiarizarse con la programación que se había estado manejando en el animatronic y de esta manera poder determinar cuáles eran los datos disponibles para su monitoreo y la programación necesaria para su manipulación a larga distancia.

Una vez comprendido esto procedí a realizar pruebas de programaciones básicas en un plc individual.

Diseño del software o app: En esta etapa me vi en la necesidad de crear múltiples páginas que en primera instancia me permitieran reflejar los datos de activaciones de los sensores que se accionaban en el plc individual que se me proporciono para pruebas, antes de implementarlo en el animatronic, esto con el fin de no alterar la programación que anteriormente se encontraba en el mismo.

También se creó un programa primitivo en tia porta para estas pruebas de comunicación plc – página web.

Sincronización con el animatronic: Una vez lograda la comunicación del plc y la pagina comencé a trabajar con la programación que anteriormente se encontraba guardada en el plc.

En las pruebas solamente había colocado la lectura y la activación de 2 puertos del plc pero en el caso de la programación del animatronic ocupaban 9 salidas por lo que fue necesario nuevamente alterar la programación de la página para poder agregar las 7 salidas restantes.

Surgieron conflictos a la hora de arrancar el plc en algunas variables que distorsionaban la activación de las salidas ya que se había repetido ciertas etiquetas que eran necesarias crear nuevamente solo para la activación de las salidas.

Última revisión: Se hicieron múltiples revisiones a la programación y al enlace que permitían la conexión al plc hasta cerciorarme que el sistema respondía de la mejor manera.

Entrega final: Una vez realizado las revisiones se entregó el resultado final sin novedades, cumpliendo con las expectativas que se esperaban del mismo.

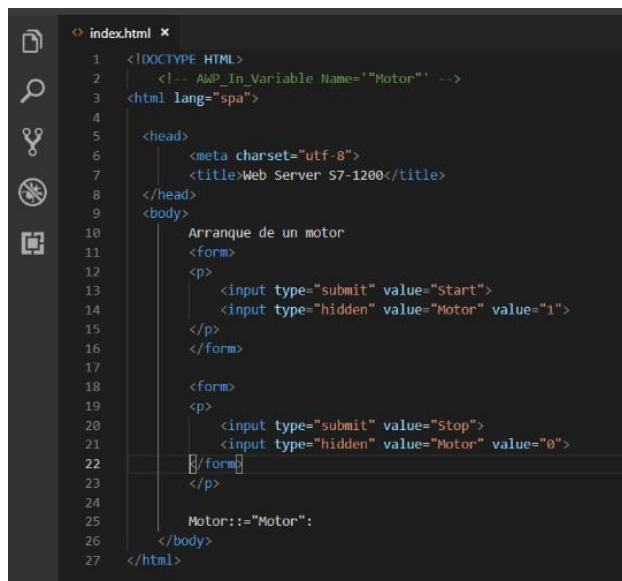
4.2 Implementación

En las primeras semanas se procedió a revisar la información bibliográfica para poder saber más acerca de las herramientas, equipos y lenguajes con los que se tenía que trabajar para poder desarrollar el proyecto planeado.

Posteriormente se procedió a la descarga e instalación del software de Siemens Tia portal v 14.0, con el cual se pretendía realizar la programación del plc que controla el Animatronic.

En la siguiente semana se estudió las propiedades del plc para poder identificar la programación necesaria para poder realizar la programación de la página por medio del lenguaje HTML.

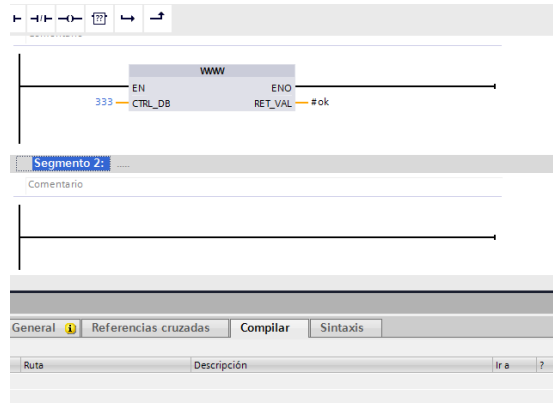
Se inició la programación de una página básica que me permitiera saber si la programación del plc y el lenguaje de programación en la página HTML eran compatible para poder manejar el Animatronic vía web.



```
1 <!DOCTYPE HTML>
2 <!-- AWP_In_Variable Name="Motor" -->
3 <html lang="spa">
4
5 <head>
6 <meta charset="utf-8">
7 <title>Web Server 57-1200</title>
8 </head>
9 <body>
10 Arranque de un motor
11 <form>
12 <p>
13 <input type="submit" value="Start">
14 <input type="hidden" value="Motor" value="1">
15 </p>
16 </form>
17
18 <form>
19 <p>
20 <input type="submit" value="Stop">
21 <input type="hidden" value="Motor" value="0">
22 </form>
23 </p>
24
25 Motor::="Motor":
26 </body>
27 </html>
```

El programa de usuario debe llamar la instrucción " WWW" para que la aplicación web, es decir, las páginas web definidas por el usuario, estén disponibles en las páginas web estándar de la CPU y puedan abrirse.

El DB de Web Control generado a partir de los archivos de origen es el parámetro de entrada (CTRL_DB) para la instrucción "WWW". El DB de Web Control referencia el contenido de las páginas web definidas por el usuario que está codificado en los DBs de fragmento y contiene información de estado y control.



Se empleó activo la característica "Web Server" en el plc, donde se cargó la página y se generó los bloques necesarios para poder ser compilado y cargado al plc.

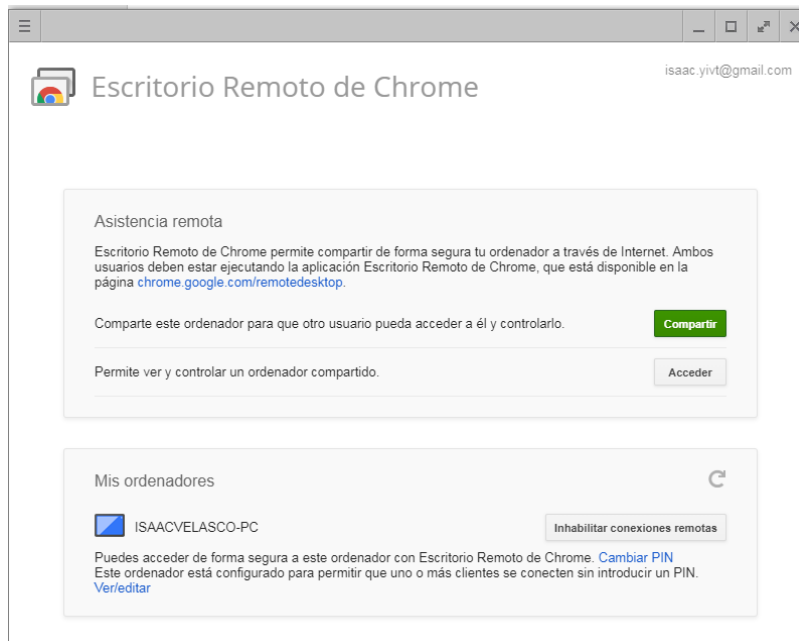
Al principio tuve detalles ya que tenía el acceso restringido para el usuario del servidor web, continúe leyendo el manual del servidor web e identifique el problema.

Cree un usuario con acceso de administrador para poder abrir la página HTML que había creado.

Desafortunadamente solo pude acceder a la página del plc por medio de una red local y en las siguientes semanas continúe revisando el manual para identificar cual era el detalle, pero no logre conseguir ningún resultado positivo.

Entonces opte por buscar más opciones que me pudieran resolver el detalle que tenía para acceder al control del plc vía web.

Encontré una aplicación que me permitía poder acceder a mi pc desde cualquier lugar mediante el servicio de Google Remote Desktop. Esto soluciono el acceso restringido que tenía a la página del plc.



Una vez resuelto esto, comencé a trabajar de lleno con la página que controlaría el plc. Nuevamente recurrí a investigar un poco más a fondo la programación necesaria para poder hacer la comunicación entre el plc y la página web. La cual me debería arrojar los datos de las activaciones de cada una de las válvulas de salidas y permitirme controlar directamente desde un dispositivo remoto al animatronic.

Surgieron algunas dudas y tuve que ocupar asesorías con el fin de resolver algunos detalles que se me presentaron a la hora de generar los bloques que permitirían la comunicación.

Ya con este conocimiento pude resolver esta problemática y concluir con la programación y el diseño de la página que permitiría de manipulación del animatronic.

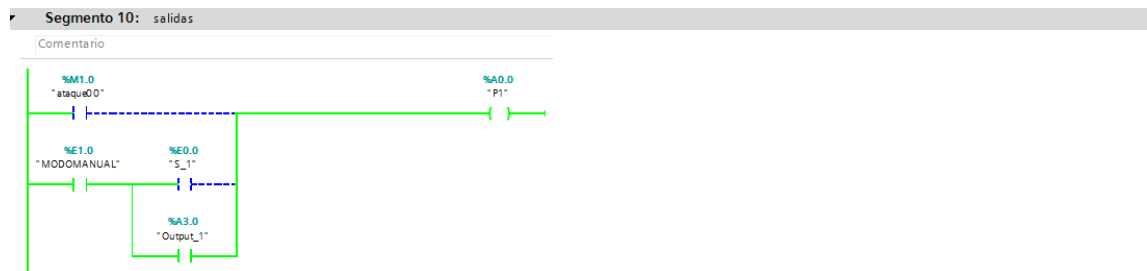
A continuación, anexo la programación del plc y de la página html.

4.2.1 Programación del PLC

Empezando por la posición inicial que se muestra a continuación en la siguiente imagen.



En las siguientes imágenes se muestra la programación que se elaboró en el plc para darle movimiento al animatronic. Como podemos observar, en cada una de ellas se encuentra la programación que acciona una parte del cuerpo robotico.

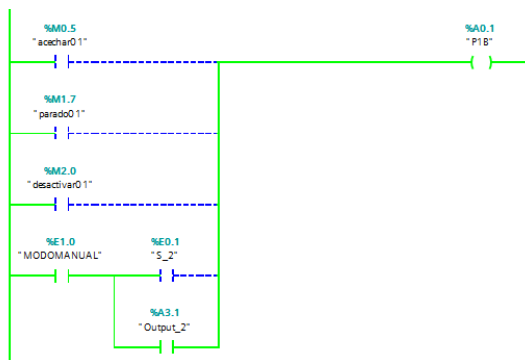


En esta etapa se encuentran cuatro variables que nos permiten como resultado el movimiento que se muestra en la siguiente imagen a continuación.



Como podemos observar, al accionar el interruptor físico asignado con la variable “S_1” se activa la salida “P1” que le permite al animatronic levantar la parte delantera de su cuerpo al estirar las patas delanteras. En la misma instancia podemos observar que a la par esta salida puede ser activada por medio de la variable “Output_1”, esta variable fue asignada para poder tener accionarlo de manera remota independientemente del interruptor físico.

En el siguiente bloque de programación de igual manera podemos ver que al accionar el interruptor físico asignado con la variable “S_2” se activa la salida “P1B” que le permite al animatronic levantar la parte inferior de su cuerpo al estirar las patas traseras.

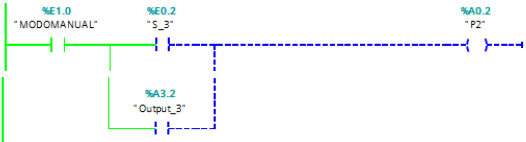


De igual manera se le asigno la variable “Output_2” que nos permite controlarlo remotamente.

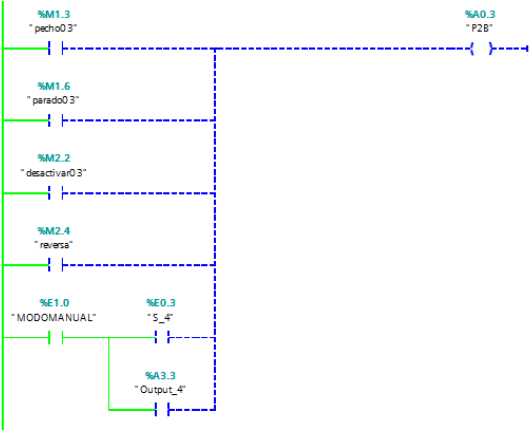
En la siguiente imagen podemos observar el movimiento que obtenemos como resultado.

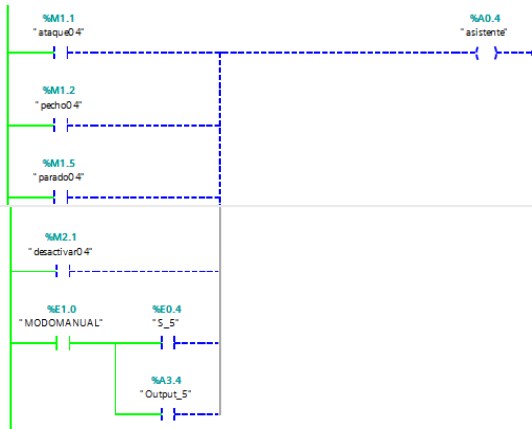


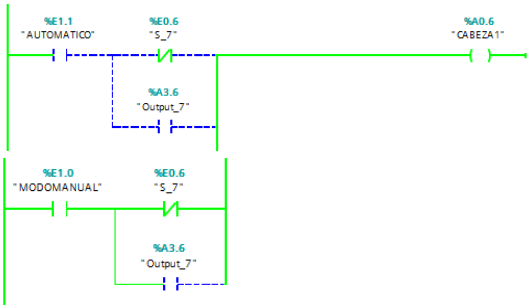
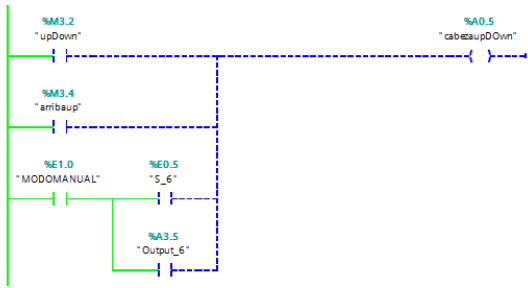
En los próximos bloques de igual manera se mostrarán los bloques de programaciones que fueron necesarios para dar movimiento al animatronic seguido de la acción que se logró por consecuencia.

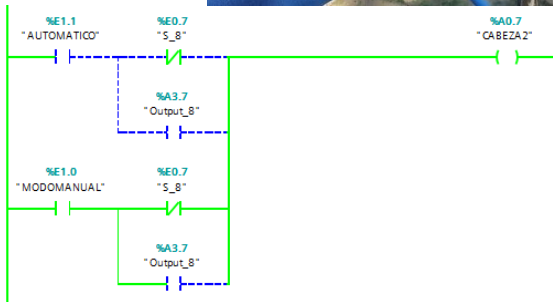
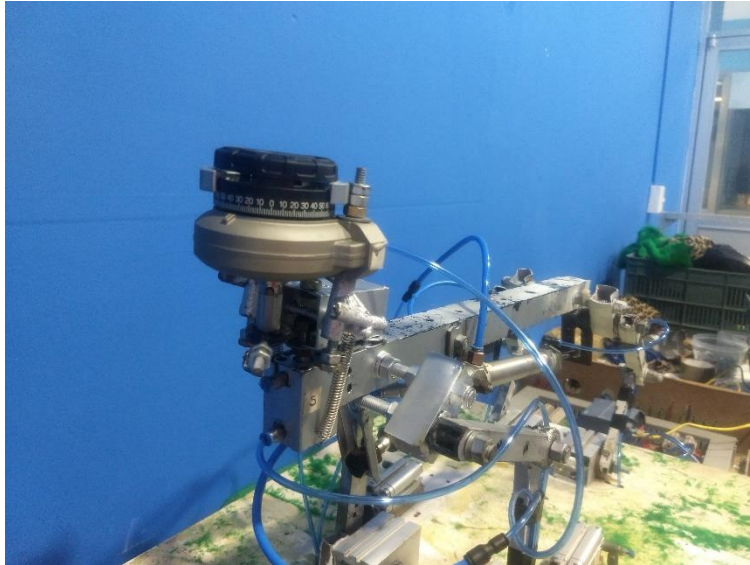


Movimiento resultante.









Estas últimas dos programaciones nos permiten mover la cabeza de un lado para otro por medio de un electro válvula rotatoria en la parte superior delantera del animatronic. Que de igual manera puede ser controlado de manera remota.

4.2.2 Programación de la página de usuario en HTML

Esta siguiente etapa se muestra la programación que fue necesaria para hacer la comunicación con el plc. En la primera etapa se inicia el htm y posterior mente se colocan los comandos AWP que nos permiten leer y escribir variables especiales en el plc. Este comando se asignó a las variables declaradas en el plc que fueron asignadas para el accionamiento remoto las cuales se muestran a continuación.

```
<html>
<!-- AWP_In_Variable Name="Output_1" -->
<!-- AWP_In_Variable Name="Output_2" -->
<!-- AWP_In_Variable Name="Output_3" -->
<!-- AWP_In_Variable Name="Output_4" -->
<!-- AWP_In_Variable Name="Output_5" -->
<!-- AWP_In_Variable Name="Output_6" -->
<!-- AWP_In_Variable Name="Output_7" -->
<!-- AWP_In_Variable Name="Output_8" -->
<!-- AWP_In_Variable Name="Output_9" -->
<!-- AWP_In_Variable Name="P1" -->
<!-- AWP_In_Variable Name="P1B" -->
<!-- AWP_In_Variable Name="P2" -->
<!-- AWP_In_Variable Name="P2B" -->
<!-- AWP_In_Variable Name="asistente" -->
<!-- AWP_In_Variable Name="cabezaupDOWn" -->
<!-- AWP_In_Variable Name="CABEZA1" -->
<!-- AWP_In_Variable Name="CABEZA2" -->
<!-- AWP_In_Variable Name="rugido" -->
```

En este siguiente bloque se colocó los colores del texto, el nombre de la página y las dimensiones de una tabla que me permitiera tener lo botones y las variables de una manera más ordenada. También se le asignó un tiempo determinado para que la página se actualizará, esto con el fin de tener el seguimiento de las activaciones.

```
<font color="black">
<head>
<meta http-equiv="refresh" content="1" />
<title>Welcome to my webpage</title>

</head>
<table class="egt" width="100%" height="100%" border=red 5px solid;>
```

```
<tr>
<body>
```

En esta etapa procedí a programar los botones que permiten activar la variable asignada para cada movimiento del animatronic. También se programó una sección en la cual podía visualizar si en realidad estaba activo esa salida asignándole el nombre de la variable que se colocó en la programación del plc para evitar confusiones.

```
<td>
<form>
  <input style="color:#FFFFFF" type="submit" value="Encender 1">
  <input type="hidden" name="Output_1" value="1">
</form>
<form>
  <input style="color:#FFFFFF" type="submit" value="Apagar 1">
  <input type="hidden" name="Output_1" value="0">
</form>
</td>
<td><font>Output_1 : :="P1":</font></td>
</body>
<body>
</tr>
```

A continuación, solamente se copió la programación anteriormente mencionada solamente alterando las variables que hacían falta por agregar a la programación para que fuese posible su activación remota. Iniciando desde la variable Output_1 hasta Output_9.

```
<tr>

<td>
<form>
  <input style="color:#FFFFFF" type="submit" value="Encender 2">
```



```
<input type="hidden" name="Output_2" value="1">
</form>
<form>
  <input style="color:#FFFFFF" type="submit" value="Apagar 2">
  <input type="hidden" name="Output_2" value="0">
</form>
</td>

<td><font>Output_2 : :="P1B":</font></td>

</tr>
</body>
<body>
</tr>

<tr>

  <td>
  <form>
    <input style="color:#FFFFFF" type="submit" value="Encender 3">
    <input type="hidden" name="Output_3" value="1">
  </form>
  <form>
    <input style="color:#FFFFFF" type="submit" value="Apagar 3">
    <input type="hidden" name="Output_3" value="0">
  </form>
</td>

  <td><font>Output_3 : :="P2":</font></td>

</tr>
</body>
<body>
</tr>

<tr>

  <td>
  <form>
    <input style="color:#FFFFFF" type="submit" value="Encender 4">
    <input type="hidden" name="Output_4" value="1">
  </form>
  <form>
    <input style="color:#FFFFFF" type="submit" value="Apagar 4">
```

```
<input type="hidden" name="Output_4" value="0">
</form>
</td>

<td><font>Output_4 : :="P2B":</font></td>

</tr>
</body>
<body>
</tr>

<tr>

<td>
<form>
<input style="color:#FFFFFF" type="submit" value="Encender 5">
<input type="hidden" name="Output_5" value="1">
</form>
<form>
<input style="color:#FFFFFF" type="submit" value="Apagar 5">
<input type="hidden" name="Output_5" value="0">
</form>
</td>

<td><font>Output_5 : :="asistente":</font></td>

</tr>
</body>
<body>
</tr>

<tr>

<td>
<form>
<input style="color:#FFFFFF" type="submit" value="Encender 6">
<input type="hidden" name="Output_6" value="1">
</form>
<form>
<input style="color:#FFFFFF" type="submit" value="Apagar 6">
<input type="hidden" name="Output_6" value="0">
</form>
</td>
```

```
<td><font>Output_6 : :="cabezaupDown":</font></td>

</tr>
</body>
<body>
</tr>

<tr>

<td>
<form>
  <input style="color:#FFFFFF" type="submit" value="Encender 7">
  <input type="hidden" name=""Output_7"" value="1">
</form>
<form>
  <input style="color:#FFFFFF" type="submit" value="Apagar 7">
  <input type="hidden" name=""Output_7"" value="0">
</form>
</td>

<td><font>Output_7 : :="CABEZA1":</font></td>

</tr>
</body>
<body>
</tr>

<tr>

<td>
<form>
  <input style="color:#FFFFFF" type="submit" value="Encender 8">
  <input type="hidden" name=""Output_8"" value="1">
</form>
<form>
  <input style="color:#FFFFFF" type="submit" value="Apagar 8">
  <input type="hidden" name=""Output_8"" value="0">
</form>
</td>

<td><font>Output_8 : :="CABEZA2":</font></td>

</tr>
</body>
<body>
```

```

</tr>

<tr>

  <td>
    <form>
      <input style="color:#FFFFFF" type="submit" value="Encender 9">
      <input type="hidden" name="Output_9" value="1">
    </form>
    <form>
      <input style="color:#FFFFFF" type="submit" value="Apagar 9">
      <input type="hidden" name="Output_9" value="0">
    </form>
  </td>

  <td><font>Output_9 : :="rugido":</font></td>

</tr>
</body>

</table>

</font>
<a href="Tempo.html" target="_blank" id="userPageLink">tempo</a>

</html>

```

Con esta programación pudimos lograr la comunicación de la página designada por el usuario y el plc, desarrollando una interfaz amigable y de fácil comprensión para el operador del mismo.

CAPITULO V

5.1 Resultados

Al final se logró controlar cada uno de los movimientos del animatronic tanto en modo manual como en automático. La página nos permite visualizar inmediatamente la activación del electro válvulas que activan cada movimiento del animatronic al instante ya que en modo automático se sigue una rutina con tiempos de activación de cada electro válvula. De este modo pudimos lograr que el operador pudiese tener visualizada cada activación y movimiento que el animatronic realice.



Este fue la página resultante de la programación anterior mente mencionada, como podemos ver los botones están organizados para una mejor comprensión de la misma para encender o apagar individualmente cada una del electro válvulas del animatronic.

Esto asociado con el control remoto de google nos permite tener el control el animatronic, así como del pc en su totalidad en caso de tener que modificar la programación solo es necesario que tanto como el servidor y el dispositivo en el que se pretende controlar el animatronic, estén conectados a la red para lograr un enlace estable y de esta manera poder hacer las modificaciones necesarias o en efecto controlar al animatronic si así se desea.

5.2 Conclusiones y recomendaciones

Este sistema aun puede tener mejoras ya que con el tiempo pueden surgir nuevos métodos de control que permitan una mejor manipulación del animatronic y la visualización de las acciones que este realiza. Por lo pronto este sistema es eficiente y cumple con las expectativas que se tenía a la inicial el proyecto.

Se recomienda a los futuros residentes que si existe la posibilidad de optimizar los movimientos del animatronic sería de gran ayuda ya que puede observar que al acciona cada electro válvula los pistones se disparan de una manera demasiado brutal, esto a mi manera de pensar a la larga desgasta las articulaciones y los componentes del mismo al tener tal impacto al accionarlos. Esto no solo ayudaría a no desgastar la estructura ya que también le daría una presentación más profesional a la hora de exponerlo ante el público. Buscar de alguna manera poder tener un movimiento más realista del animatronic con el fin de mejorar las presentaciones y aumentar el conocimiento a los futuros ingenieros que trabajen en este prototipo.

REFERENCIAS

Wilson, Leslie B. (1993). *Comparative Programming Languages, Second Edition*. Addison-Wesley. p. 75. ISBN 0-201-56885-3. (en inglés).

Wilson, Leslie B. (1993). *Comparative Programming Languages, Second Edition*. Addison-Wesley. p. 213. ISBN 0-201-56885-3. (en inglés).

Wilson, Leslie B. (1993). *Comparative Programming Languages, Second Edition*. Addison-Wesley. p. 244. ISBN 0-201-56885-3. (en inglés).

Fondevila Gascón, Joan Francesc (2009). El peso de la televisión en el triple play de los operadores de cable en España y en Europa. *ZER, Revista de Estudios de comunicación (Journal of Communication Studies)*, 14 (27), pp. 13-31. ISSN 1137-1102. Edición digital en la Universidad del País Vasco.

Torres, Álvaro. *Telecomunicaciones y telemática. De las señales de humo a las redes de información y a las actividades por internet*. Tercera edición:2007, Colombia, Colección Telecomunicaciones.

Huidobro Moya, José Manuel. *Redes y servicios de telecomunicaciones*. Madrid: Thomson, 2006.

Huidobro Moya, José Manuel. *Tecnologías de telecomunicaciones*. México, D. F.: Alfaomega, c2006.

Herrera Pérez, Enrique. *Introducción a las telecomunicaciones modernas*. México: Limusa, 2004.

Educ.ar. (2016). ¿Qué es la robótica?. [online] Available at: <https://web.archive.org/web/20160904123705/http://www.educ.ar/sitios/educar/blogs/ver?id=122836> [Accessed 31 May 2016].

Astudillo, C. & López, H. (2012). *El cine y la tecnología: Animatronics*. Licenciatura. Escuela Politécnica Nacional.

Conejo, D. (2009). *Robots a escena! Visiones sobre la tecnología*. Máster. Universidad de Barcelona.

«What IS a PLC?» (en inglés). Consultado el 23 de enero de 2013.

M. A. Laughton, D. J. Warne (ed), Electrical Engineer's Reference book, 16th edition, Newnes, 2003 Chapter 16 Programmable Controller

Alison Dunn (12 de septiembre de 2008). «The father of invention: Dick Morley looks back on the 40th anniversary of the PLC» (en inglés). Consultado el 23 de enero de 2013.

Gútiez, I. (2017 de noviembre de 2017). Step 7 ... AWL, FUP, KOP Cuál elijo? Obtenido de PROGRAMACIONSIEMENS.COM:
<http://programacionsiemens.com/step-7-awl-fup-kop-cual-elijo/>

http://www.softwarepreservation.org/projects/FORTRAN/index.html#By_FORTRAN_project_members