Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Manual de Usuario
Instalación y Configuración

Implementación de una VPN con el micro controlador Rabbit para acceso a una red de cámaras web

Israel Alonso Figueroa – 207206463
Ing. Computación

Trimestre lectivo: 12-O

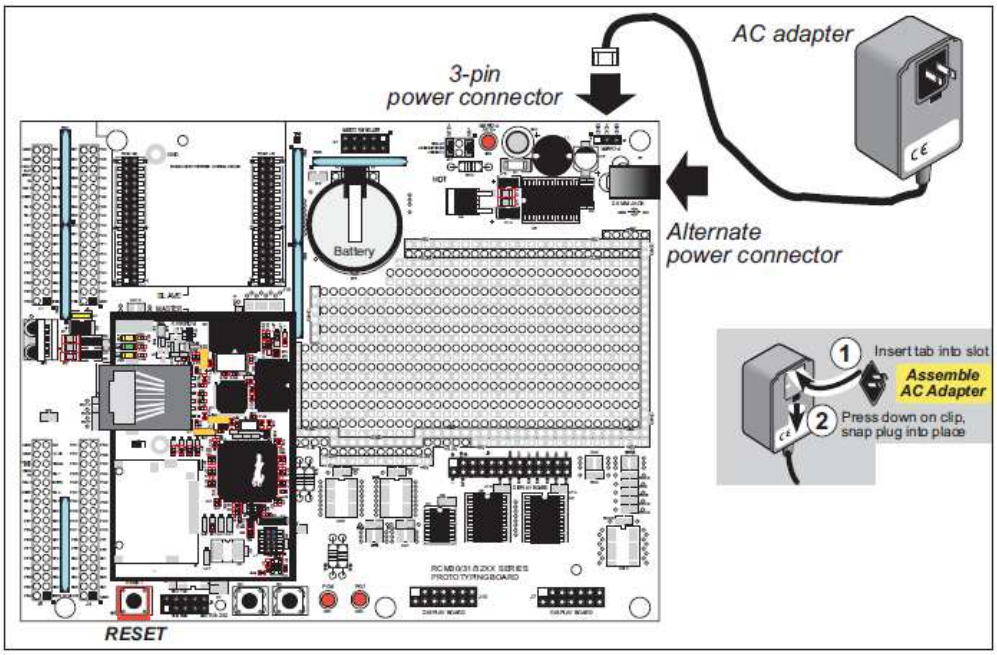
Asesor:

José Ignacio Vega Luna
Departamento: Electrónica

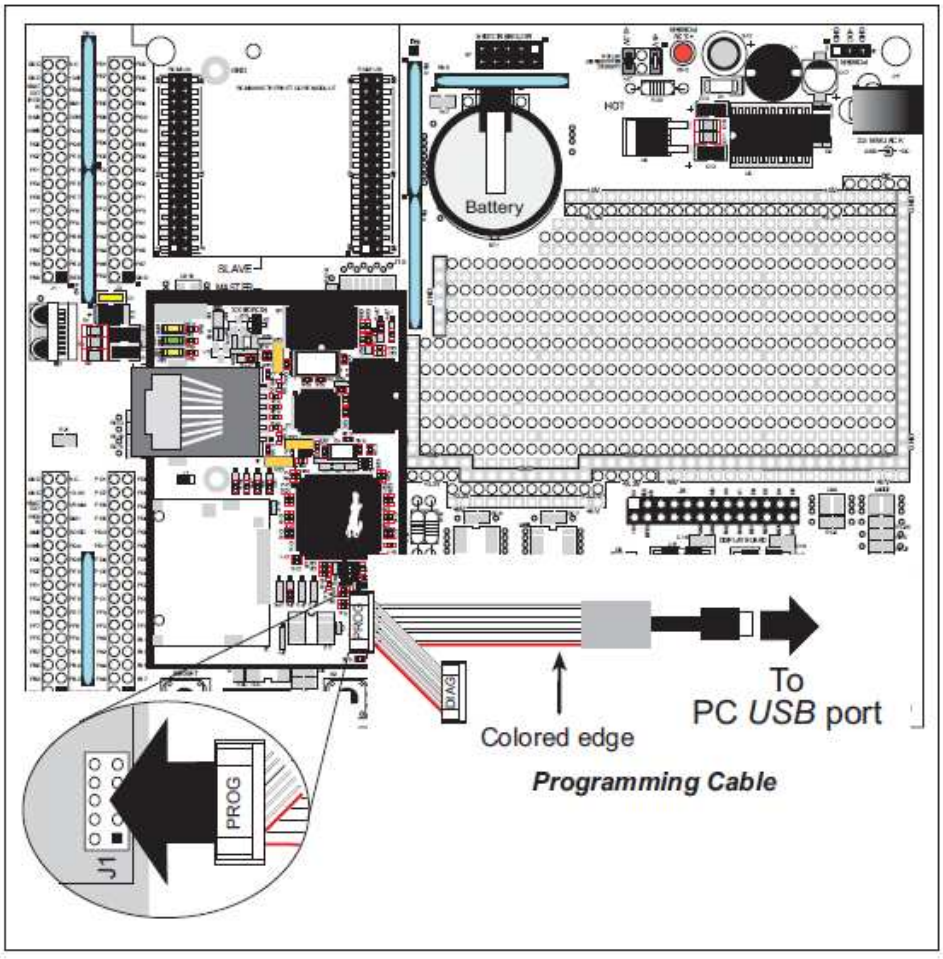
Tabla de Contenido

Conexión Tarjeta de Programación

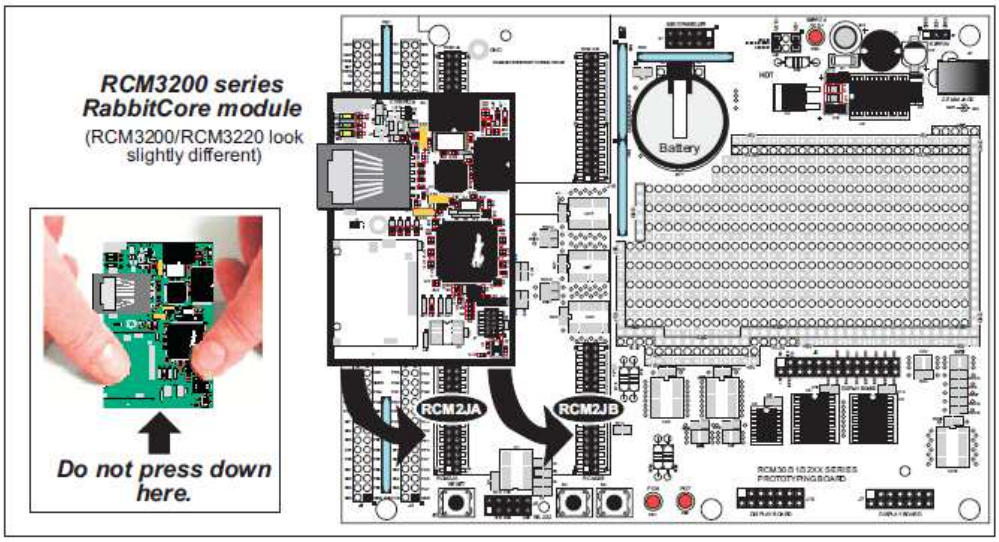
La siguiente imagen nos ilustra el cómo tiene que conectar la tarjeta para programar los micro controladores, conectamos el adaptador de corriente como se indica.



El siguiente paso es conectar el cable de programación que va del micro controlador a la Pc en el puerto Com1 (seria)

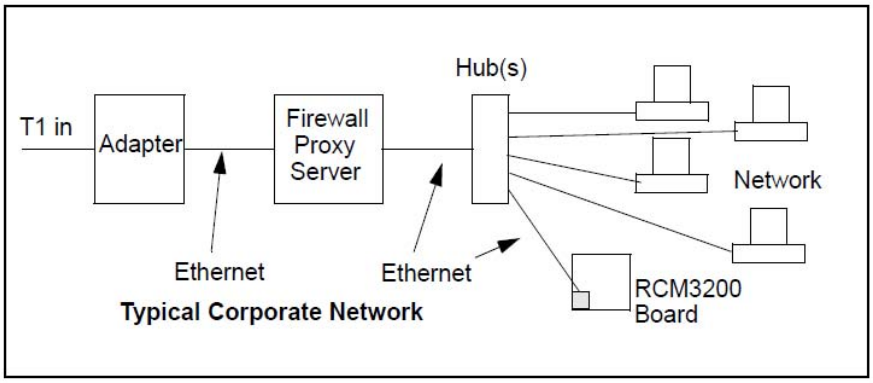


Como siguiente paso conectamos nuestro modulo micro controlador Rabbit en el bloque maestro y otro en el bloque del Esclavo como se ilustra en la imagen



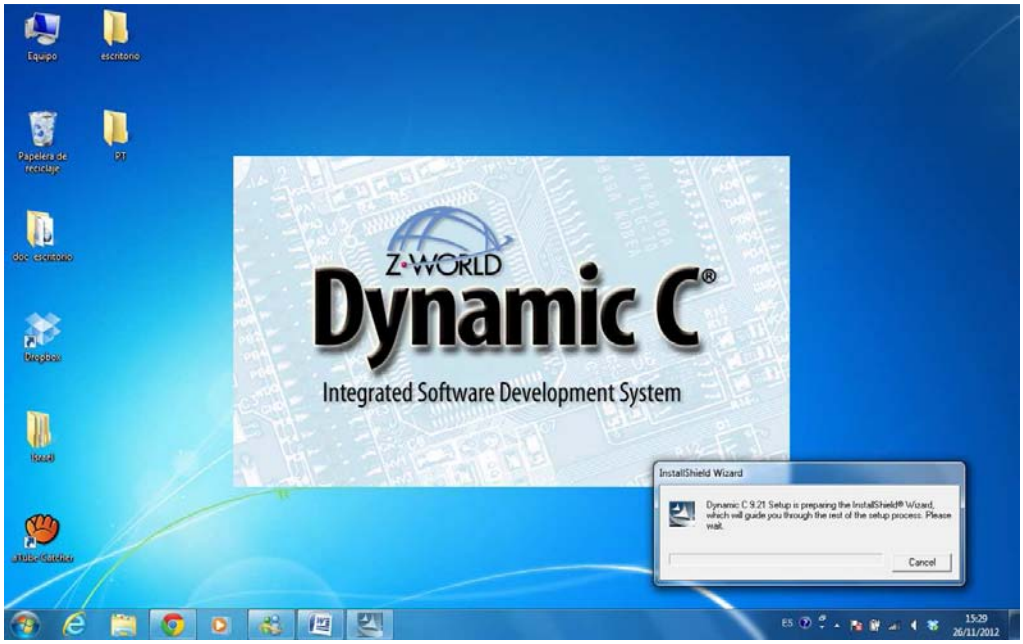
Configuración de la conexión PC- Tarjeta de Programación

Ahora mostraremos la coneccion generar en que tenemos que conetar nuestros modulos Rabbit para poder acceder a la a plicacion que instalaremos en los modulos, la coneccion se realiza por medio de un hub de red y 2 cables de red uno por modulo Rabbit

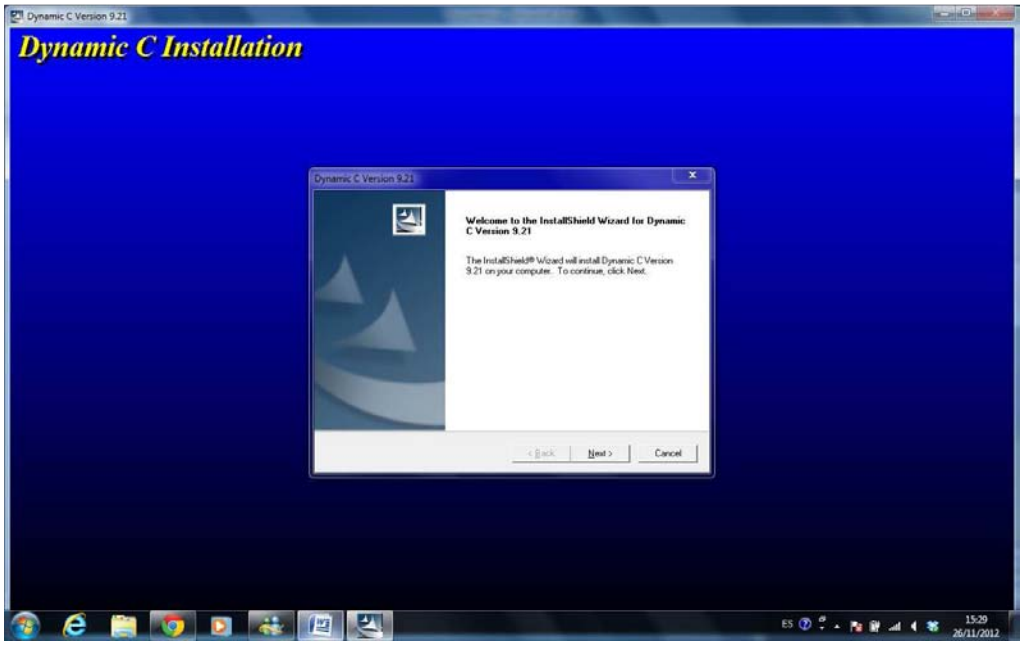


Instalación y Configuración Software Dynamic C

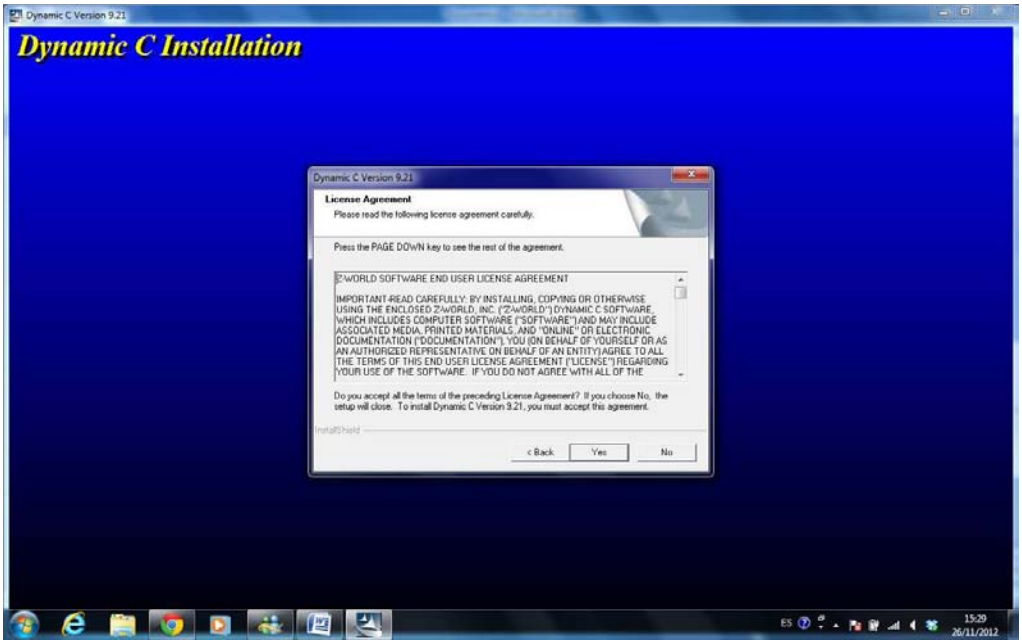
Para la instalación y configuración de nuestra aplicación lo primero que debemos hacer es realizar la instalación de nuestro software, para ello introducimos nuestro CD de Dynamic C y empezamos con la instalación como se ve en la siguiente imagen



La siguiente venta nos muestra un mensaje de bienvenida solo damos click en el botón de Next



La siguiente venta nos mostrara el contrato lo leemos y damos en el botón de Yes

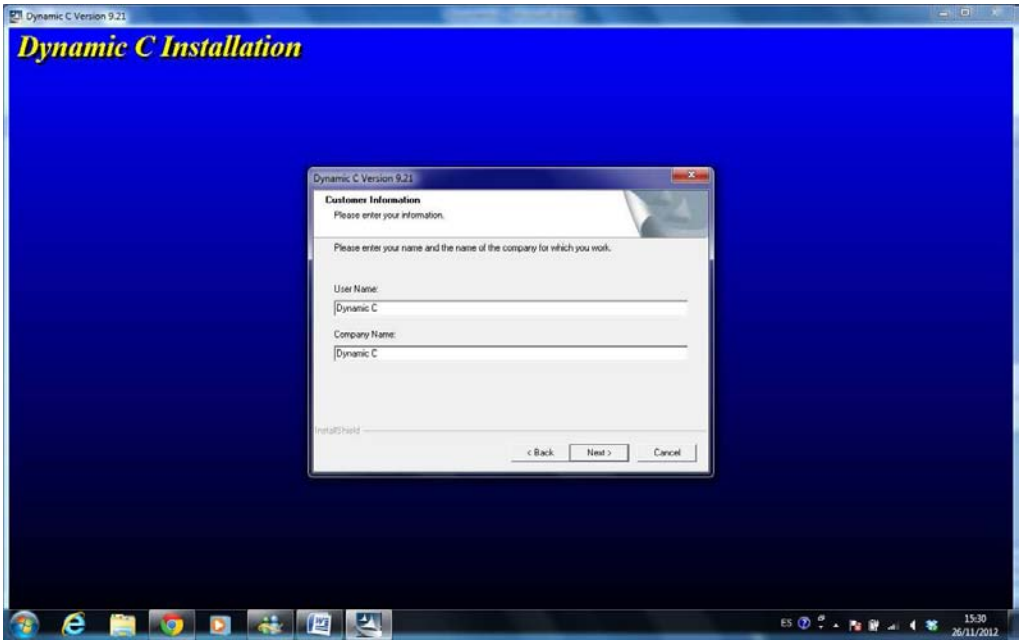


Como siguiente paso nos pedirá el Usuario y la compañía del Usuario, ingresamos los siguientes datos:

Usuario: Dynamic C

Compañía: Dynamic C

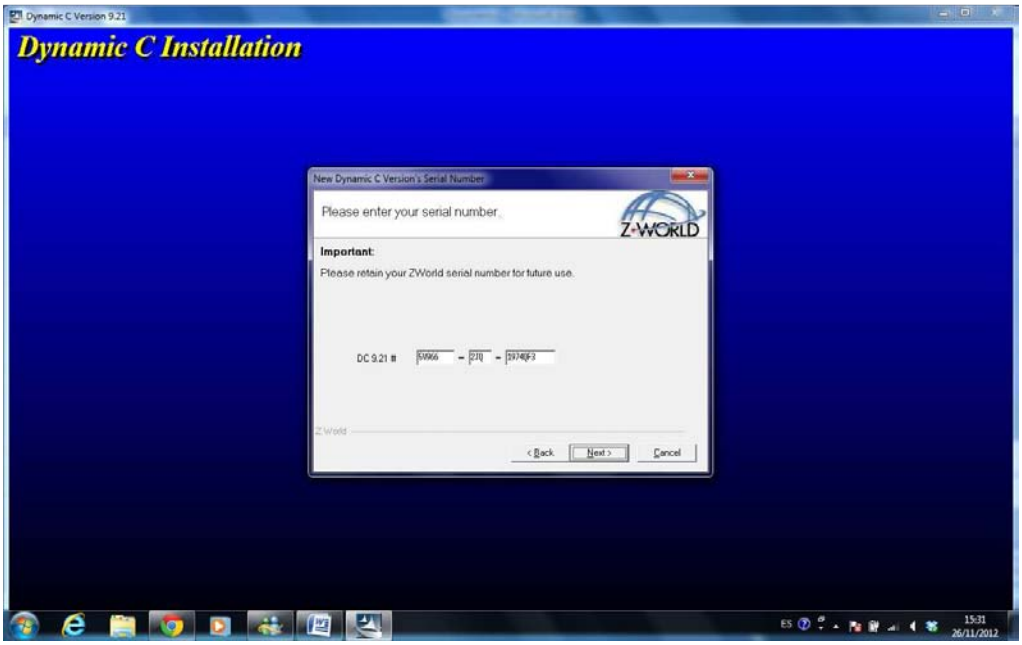
Damos click en el botón de Next.



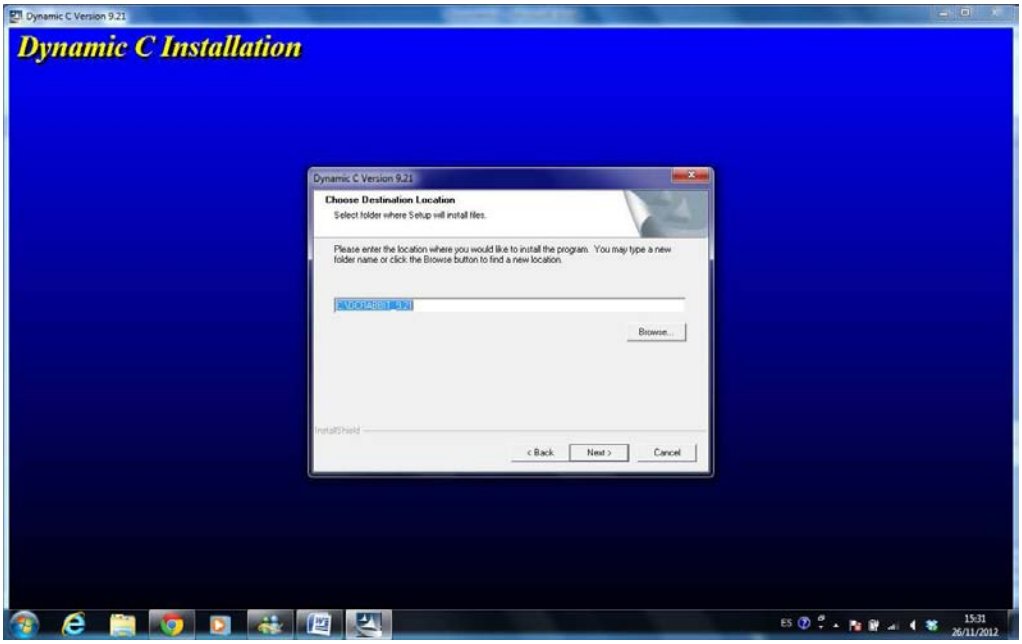
El siguiente pasó el ingreso del número de Serie del Software Dynamic C.

El número de serie: 5V966-2JQ-1974QF3

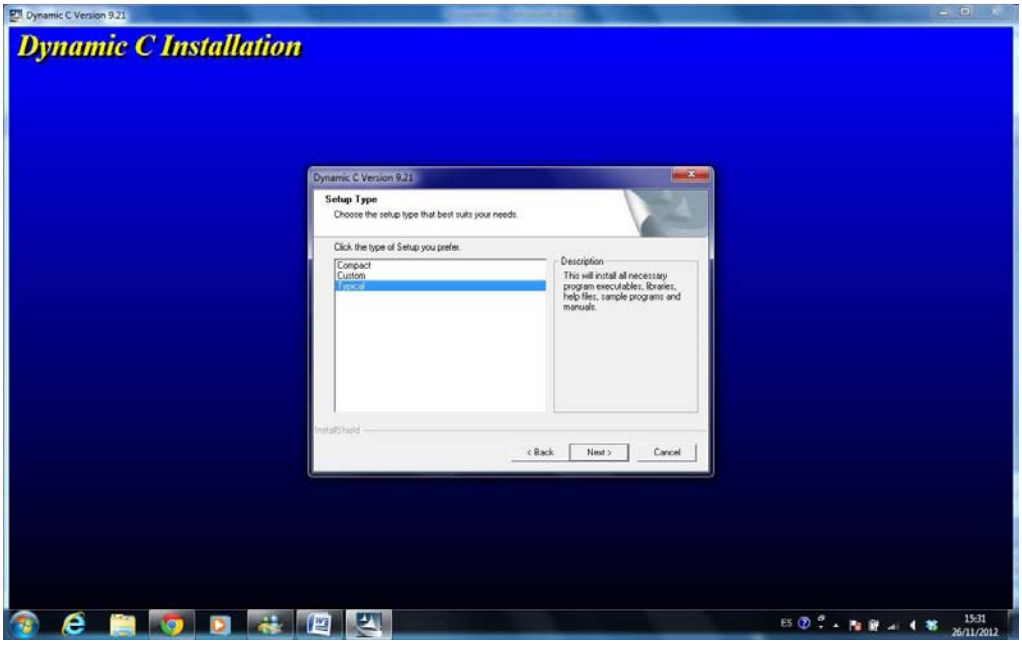
Damos clic en Next.



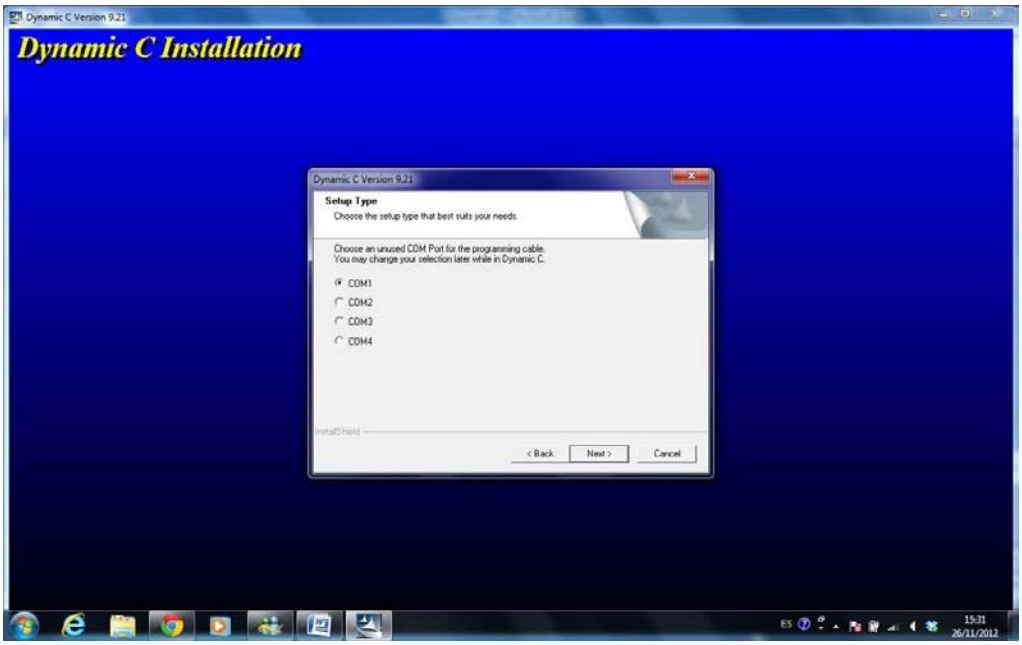
El siguiente paso es asignarle un directorio a nuestra instalación, lo dejaremos por default y pulsamos en Next.



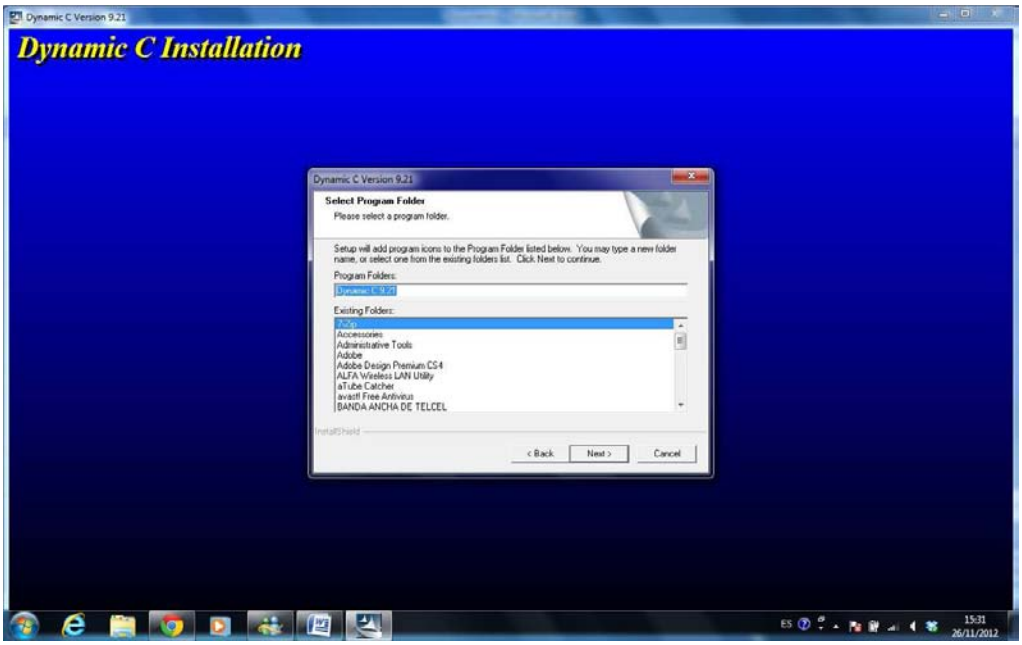
El tipo de instalación lo aremos como típica y pulsamos en Next.



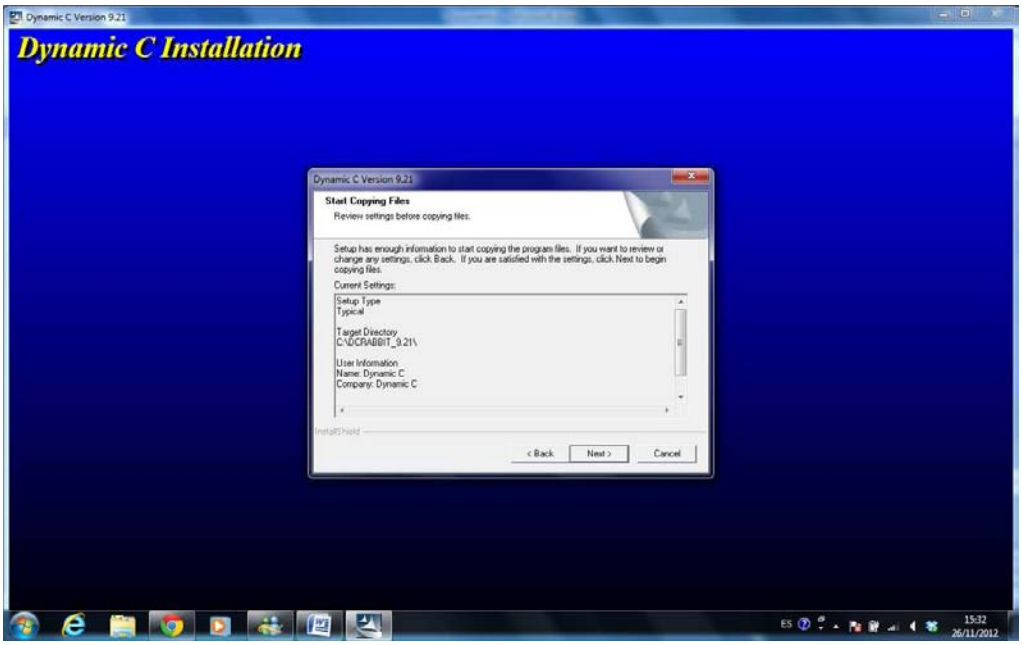
Asignamos el puerto para poder programar nuestro micro controlador Rabbit.



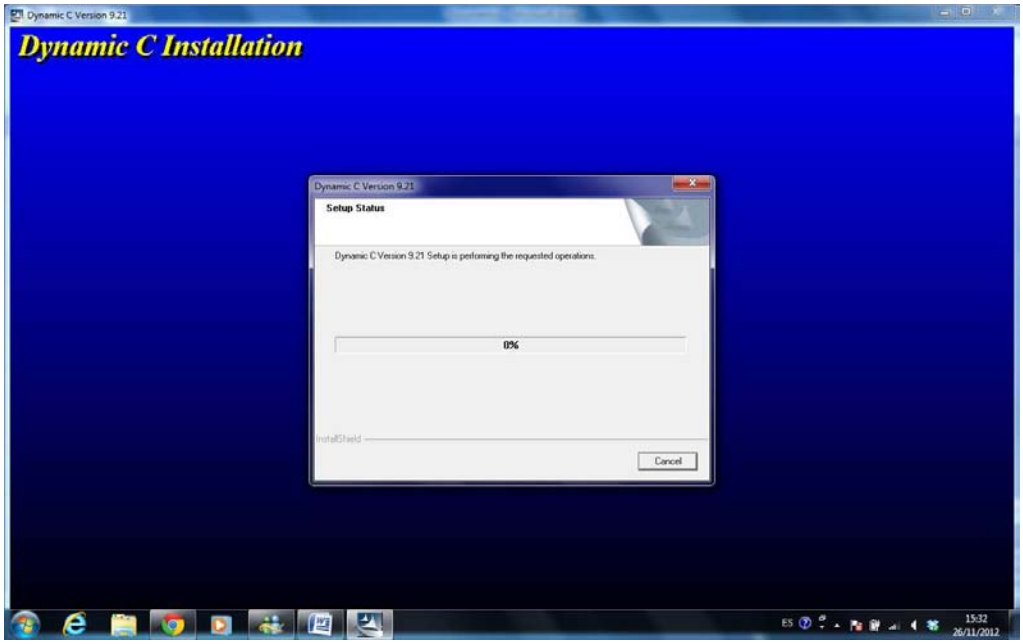
El siguiente paso nos pide que asignemos un programa para poder extraer los archivos comprimidos.



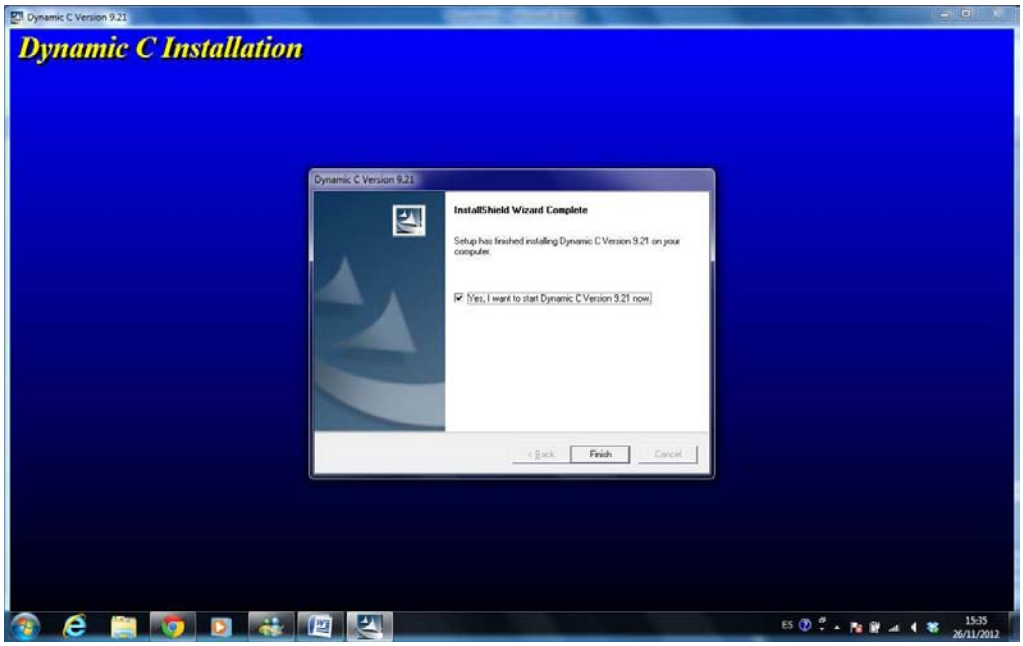
Para iniciar la instalación pulsamos en Next.



Vemos como empieza la instalación, esperamos a que finalice sin errores.

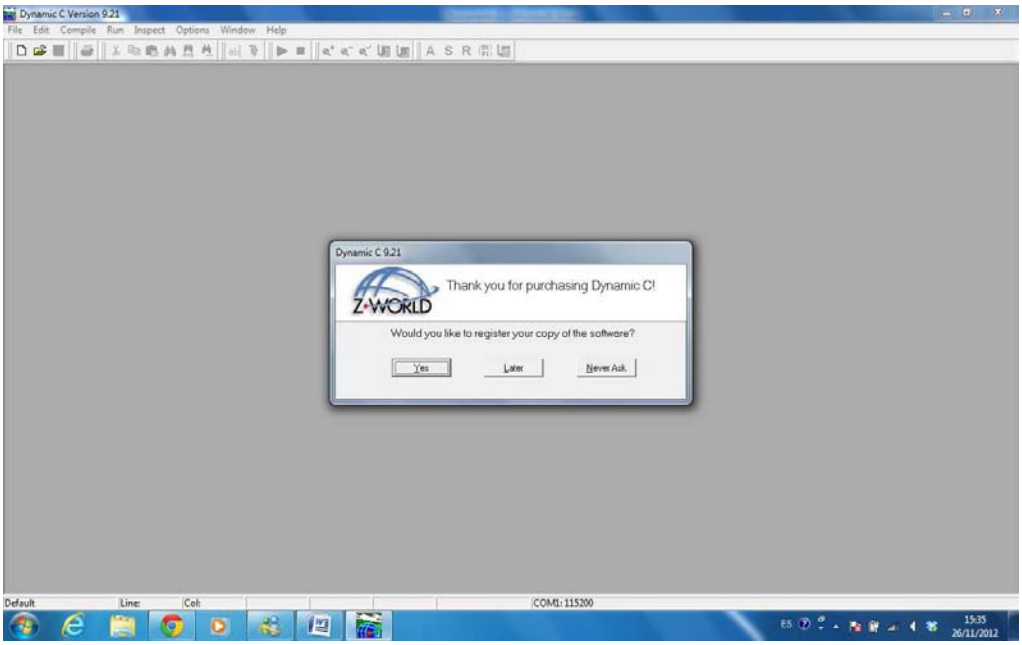


Seleccionamos la casilla de apertura del Software ya instalado y damos en Finalizar.

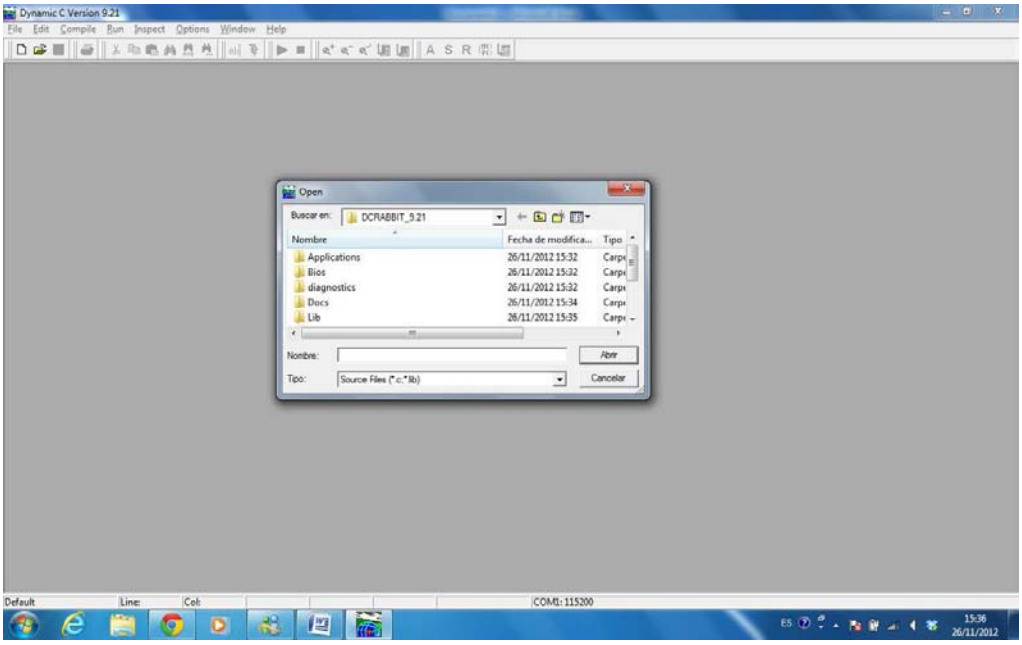


Manual Dynamic C y carga de datos

Esta es la primer pantalla de nuestro Software para la carga y programación de nuestro micro controladores Rabbit.

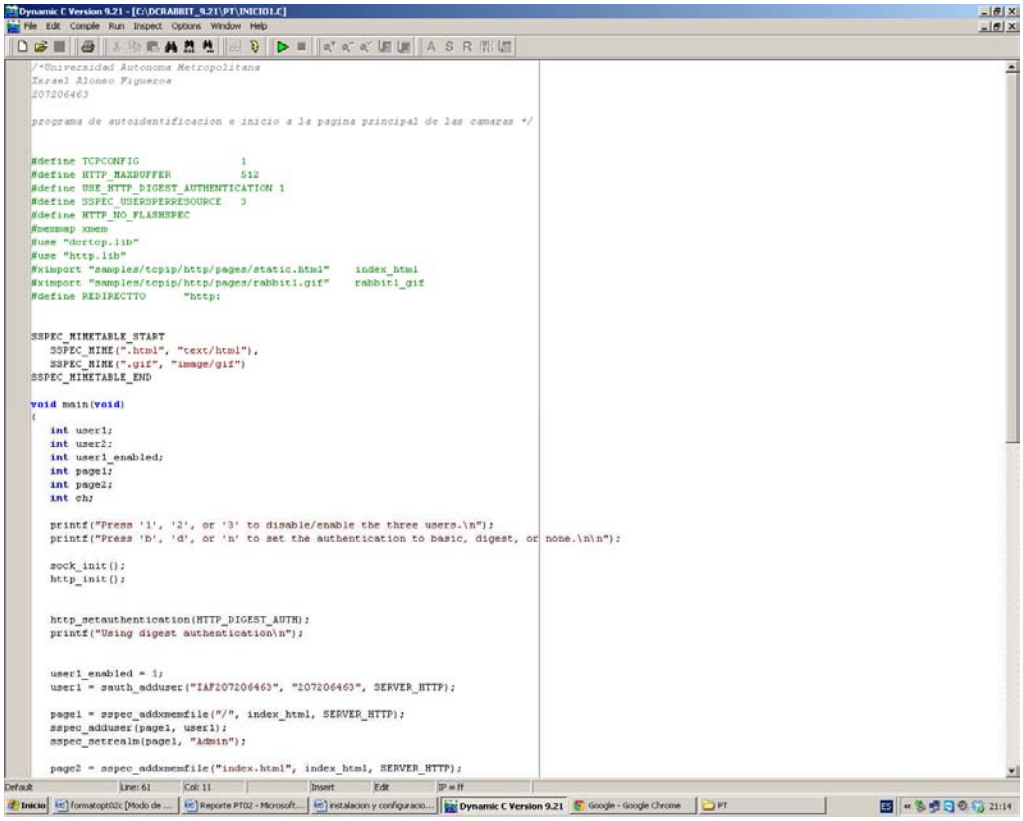


Para poder acceder a nuestro programa para el Proyecto descrito, asemos click en Archivo -> Abrir, y exploramos la carpeta para seleccionar nuestro archivo. *Inicio.c*



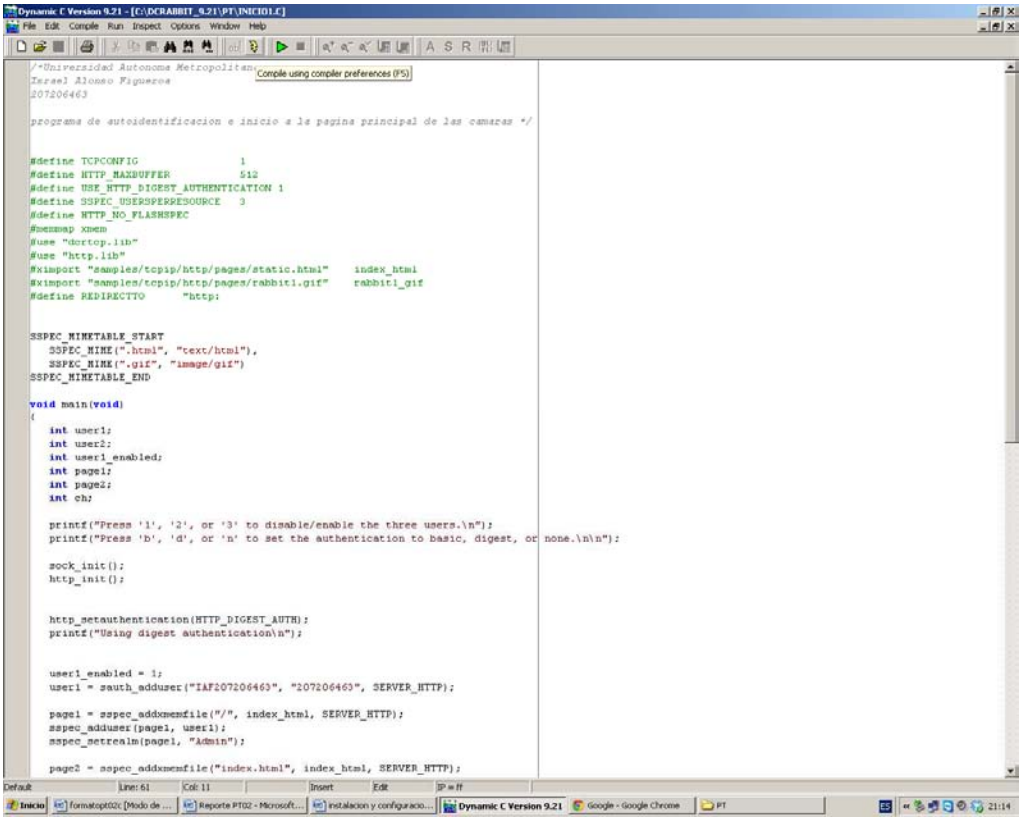
Edición de código principal

Veremos nuestro código para el proyecto.



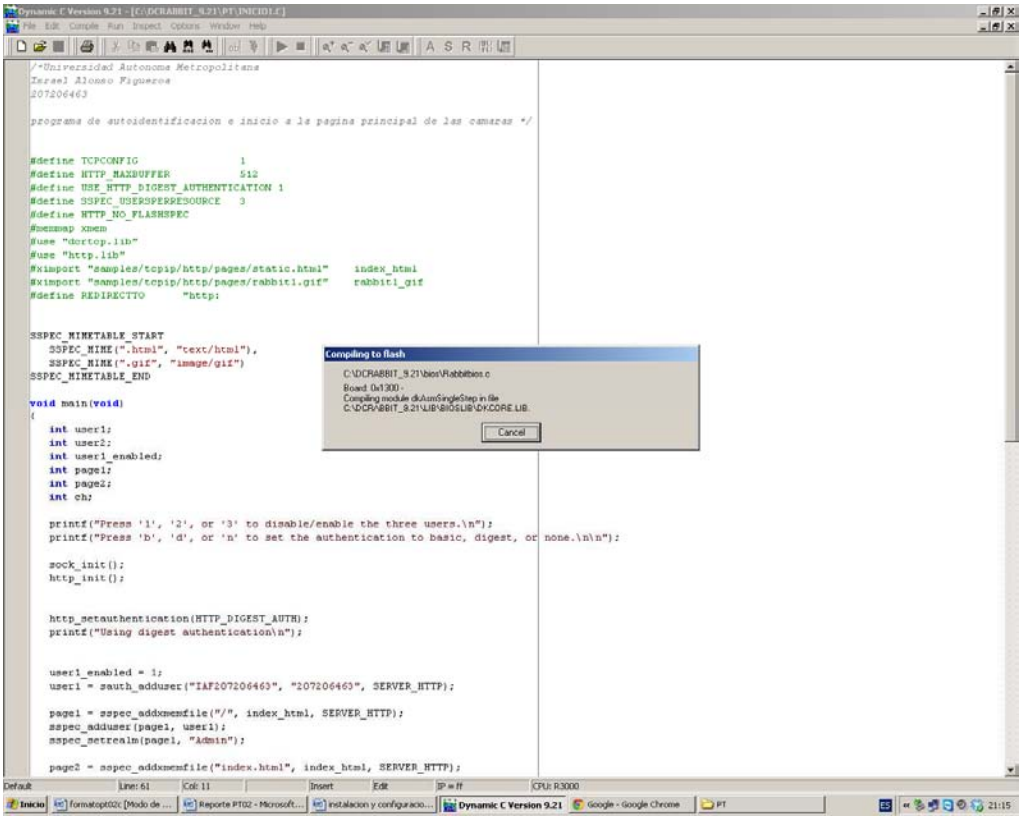
Para compilarlo y cargarlo en el micro controlador lo que tenemos que hacer es dar click en el icono del rayo y comenzara la carga del programa al micro controlador Rabbit.

Edición de Librerías de Red



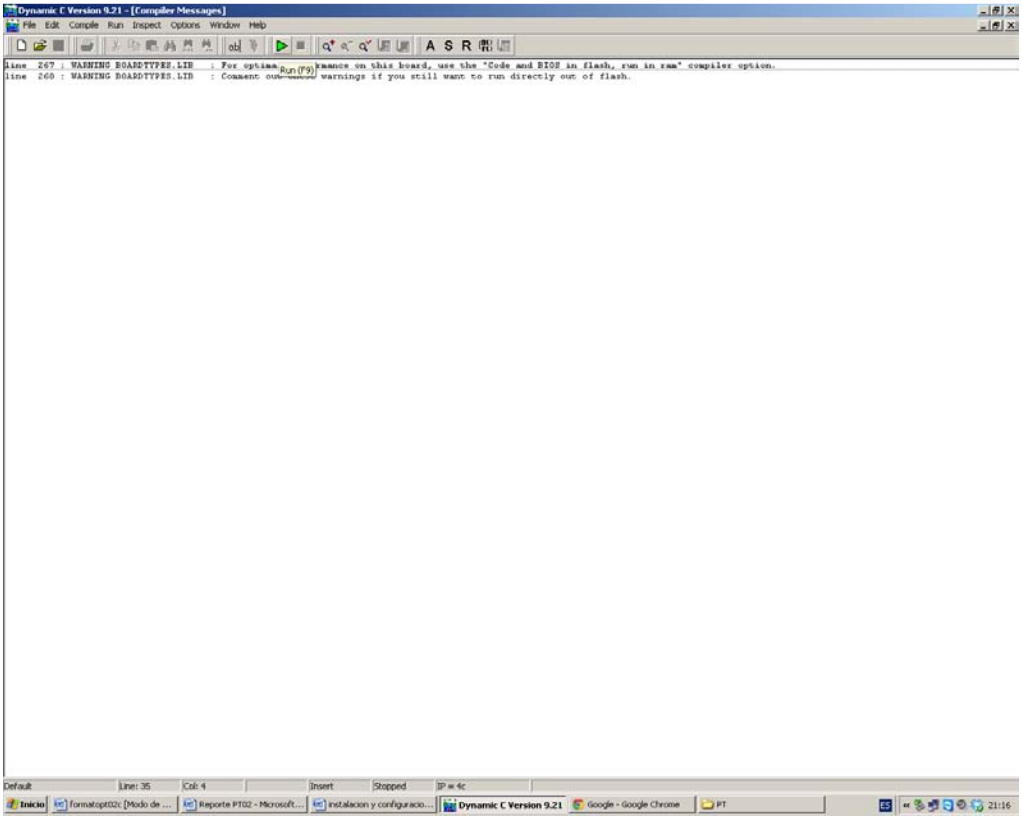
Carga de Programación a módulos Rabbit

Vemos la carga del programa al modulo Rabbit.



Inicialización del programa

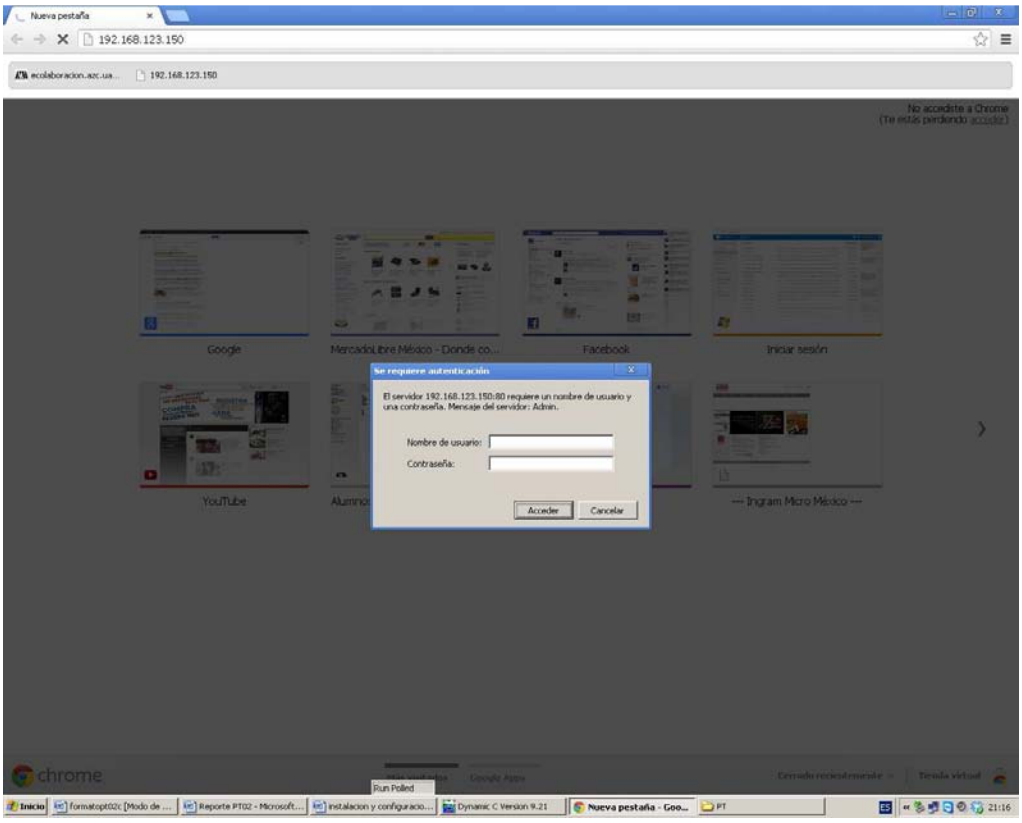
Para iniciar el programa solo presionamos F9.



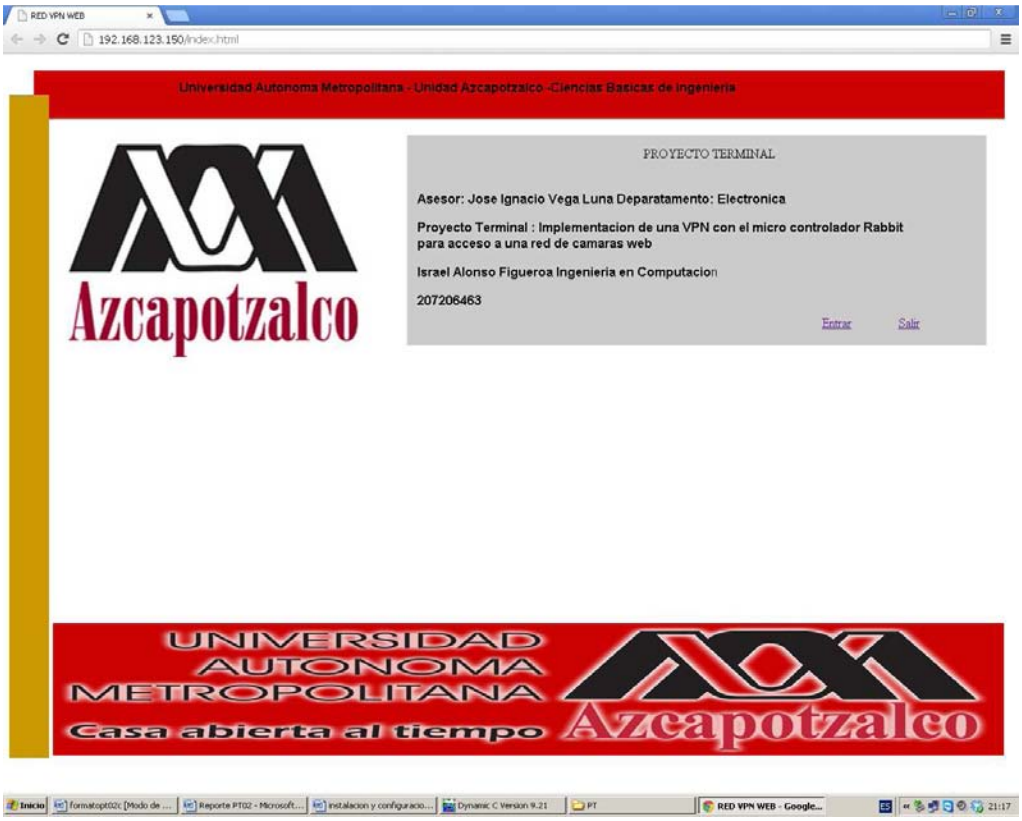
Pantallas de Prueba

La siguiente imagen ya es el progrma corriendo accediendo desde nuestro navegador de internet tecleando la direccion: 192.168.123.150 que fue editada en la librería de *tcp_config.*

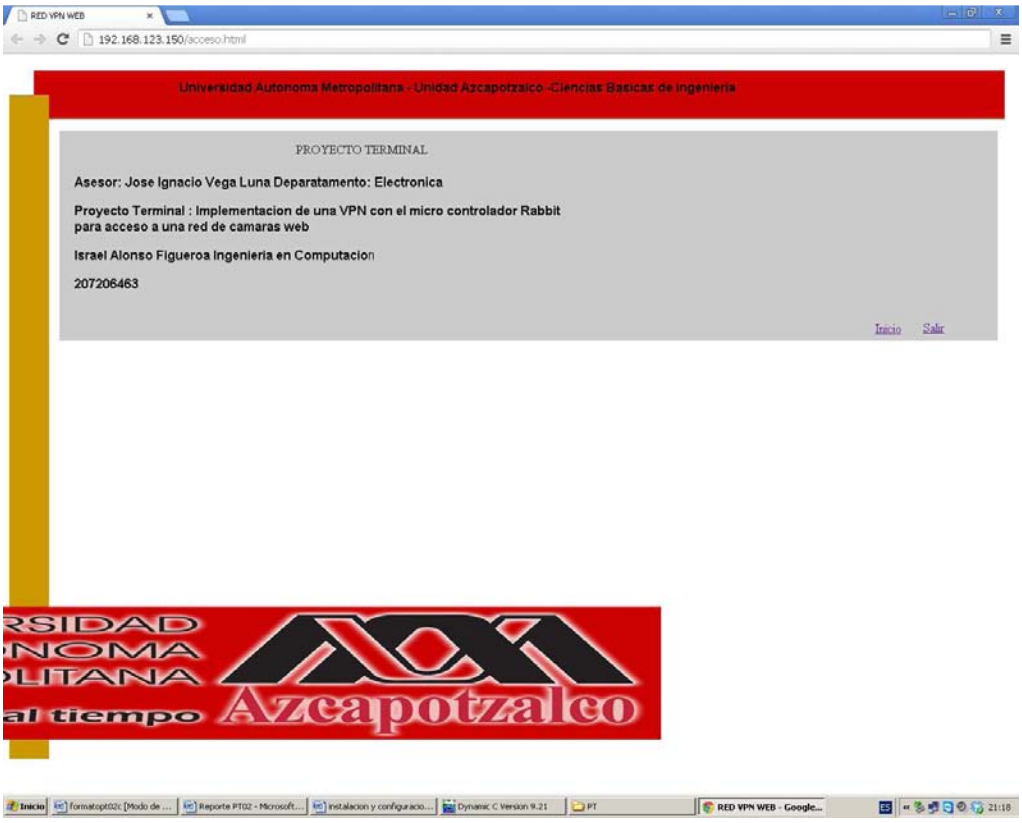
El ususario : IAF207206463
Contraseña : 207206463



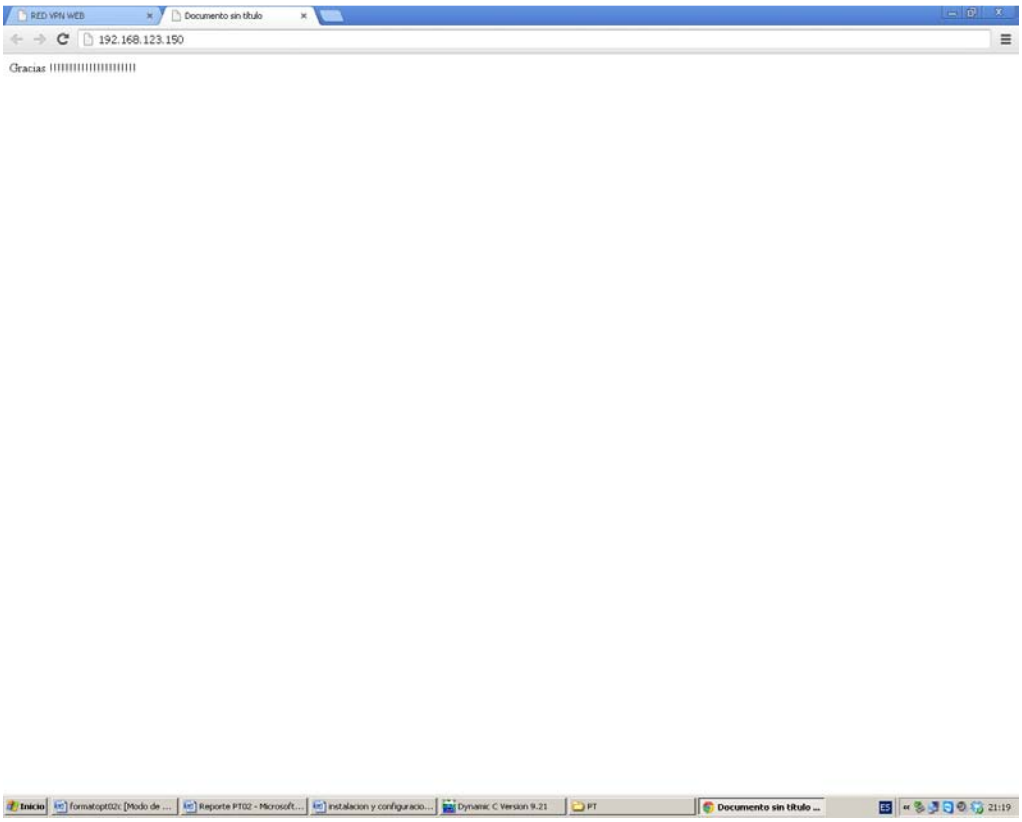
Cuando ingresemos veremos la siguiente pantalla de bienvenida, y dar click en Entrar para poder acceder a las cámaras.



Esta es la pantalla de acceso donde vernos las cámaras web.

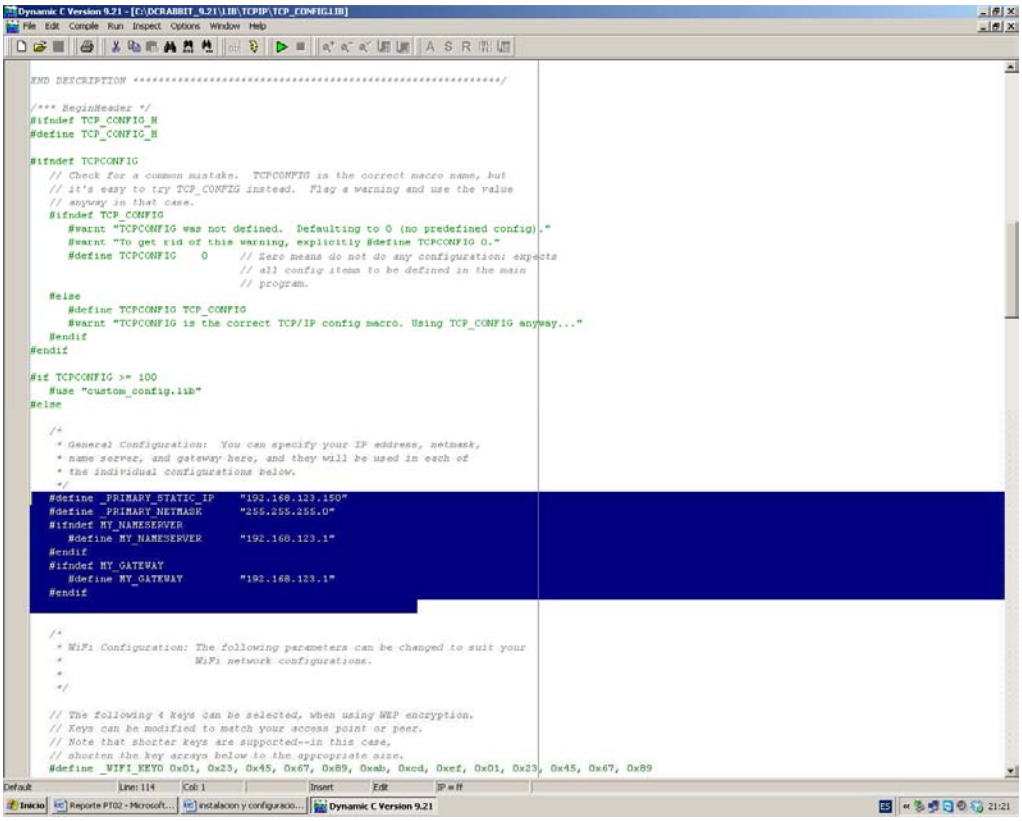


Para salir solo presionamos en el botón de Salir y nos presentara la siguiente pantalla.



Biblioteca tcp_config

Para poder cambiar la ip de acceso a nuestro modulo de acceso a la red de cámaras tenemos que editar la ip del modulo, para ello accedemos al la librería tcp_conf para editar los campos sombreados.



Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte de proyecto terminal 02

Implementación de una VPN con el micro controlador Rabbit para acceso a una red de cámaras web

Israel Alonso Figueroa – 207206463
Ing. Computación

Trimestre lectivo: 12-O

Asesor:

José Ignacio Vega Luna
Departamento: Electrónica

Tabla de contenido

Contenido

Objetivos.

Diseñar e implementar una red privada virtual VPN[1] con un micro controlador Rabbit 3000[2] con su tarjeta de desarrollo RCM3200 para acceso remoto de forma segura a una red de cámaras web instalada del Departamento de Electrónica de la UAM-Azcapotzalco.

Diseñar e implantar el protocolo a usar en la VPN.

Lo que hemos realizado en este punto es la investigación del protocolo a usar que en su caso es el IPSec característico de una Red Privada Virtual.

El IPSec es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos

Existen dos tipos de funcionamiento que en nuestro proyecto tomaremos el modo túnel: todo el paquete IP (datos más cabeceras del mensaje) es cifrado y/o autenticado. Debe ser entonces encapsulado en un nuevo paquete IP para que funcione el enrutamiento. El modo túnel se utiliza para comunicaciones red a red (túneles seguros entre routers, para VPN`s o comunicaciones ordenador a red u ordenador a ordenador sobre Internet. El propósito de este modo es establecer una comunicación segura entre dos redes remotas sobre un canal inseguro. Este ejemplo ilustra esto:

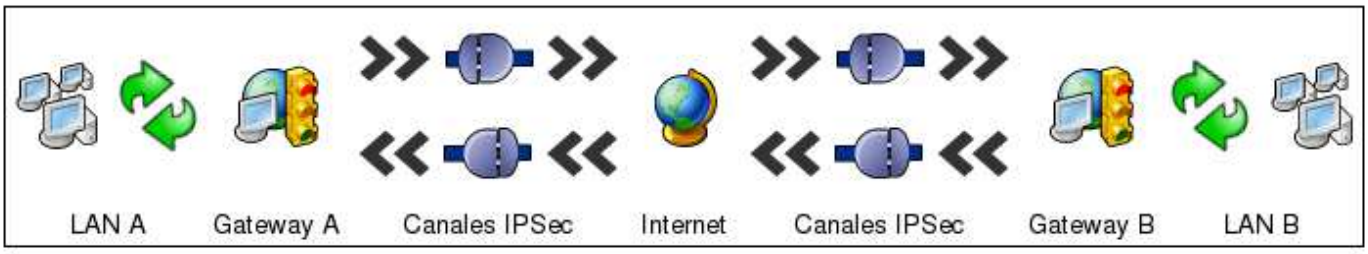
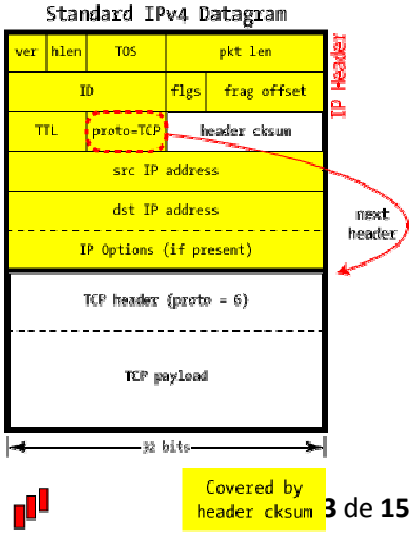


Figura 2 diagrama de conexión para el protocolo IPsec

Figura 1: Protocolo IPsec

Cabecera Ip

La cabecera del paquete IP es la siguiente:



Donde:

3.1 **ver** Es la versión del protocolo IP. IPsec se monta sobre IPv4.

3.2 **hlen** Longitud de la cabecera, en palabras de 32 bits. Su valor mínimo es de 5 para una cabecera correcta, y el máximo de 15. El tamaño de la cabecera **nunca** incluye el tamaño del payload o de la cabecera siguiente.

Indica una serie de parámetros sobre la calidad de servicio deseada durante el tránsito por una red. Algunas redes ofrecen prioridades de servicios, considerando determinado tipo de paquetes "más importantes" que otros (en particular estas redes solo admiten los paquetes con prioridad alta en momentos de sobrecarga).

3.3 **pkt len** Es el tamaño total, en octetos, del datagrama, incluyendo el tamaño de la cabecera y el de los datos. El tamaño máximo de los datagramas usados normalmente es de 576 octetos (64 de cabeceras y 512 de datos). Una máquina no debería enviar datagramas mayores a no ser que tenga la certeza de que van a ser aceptados por la máquina destino.

En caso de fragmentación este campo contendrá el tamaño del fragmento, no el del datagrama original.

3.4 **ID** Indica el identificador del fragmento actual en caso de que el paquete estuviera fragmentado

3.5 **fgls** Actualmente utilizado sólo para especificar valores relativos a la fragmentación de paquetes:

```
bit 0: Reservado; debe ser 0
  bit 1: 0 = Divisible, 1 = No Divisible (DF)
  bit 2: 0 = Último Fragmento, 1 = Fragmento Intermedio (le siguen más fragmentos) (MF)
```

La indicación de que un paquete es indivisible debe ser tenida en cuenta bajo cualquier circunstancia. Si el paquete necesitara ser fragmentado, no se enviará.

3.6 **frag offset** En paquetes fragmentados indica la posición, en unidades de 64 bits, que ocupa el paquete actual dentro del datagrama original. El primer paquete de una serie de fragmentos contendrá en este campo el valor 0.

3.7 **TTL** Indica el máximo número de enrutadores que un paquete puede atravesar. Cada vez que algún nodo procesa este paquete disminuye su valor en uno por cada Router que pase. Cuando llegue a ser 0, el paquete no será reenviado.

3.8 **protocolo** Indica el protocolo de siguiente nivel utilizado en la parte de datos del datagrama. Los más utilizados son:

Código	Descripción
1	ICMP — Internet Control Message Protocol
2	IGMP — Internet Group Management Protocol
4	IP en IP (una encapsulación IP)
6	TCP — Transmisión Control Protocolo
17	UDP — User Datagram Protocolo
41	IPv6 — next-generation TCP/IP
47	GRE — Generic Router Encapsulation (usado por PPTP)
50	IPsec: ESP — Encapsulating Security Payload
51	IPsec: AH — Authentication Header

Figura 3 diagrama donde podernos ver los diferentes tipos de código para el protocolo.

Cabecera IPsec AH (authentication only)

AH está dirigido a garantizar integridad sin conexión y autenticación de los datos de origen de los datagramas IP. Para ello, calcula un Hash Message Authentication Code (HMAC) a través de algún algoritmo hash operando sobre una clave secreta, el contenido del paquete IP y las partes inmutables del datagrama

4.1 AH en Modo Túnel

El modo túnel es el más común para dar una funcionalidad de VPN, donde un paquete IP es encapsulado dentro de otro y enviado a su destino.

Igual que en el modo transporte, el paquete es "sellado" con un ICV para autentificar al que envía la información para prevenir modificaciones durante el tránsito. Pero a diferencia del modo de transporte, el modo túnel encapsula todo el paquete IP, no sólo la carga útil (TCP, UDP, etc.). Esto hace que el destinatario del paquete sea uno diferente al destinatario original. Esto ayuda a la formación de un túnel.
Cuando un paquete en modo túnel llega a su destino, pasa el mismo proceso de autentificación igual que cualquier paquete AH-IPsec. Este proceso hace que se despoje de sus cabeceras IP y AH, luego nos queda el datagrama original, que es enrutado mediante un proceso normal.

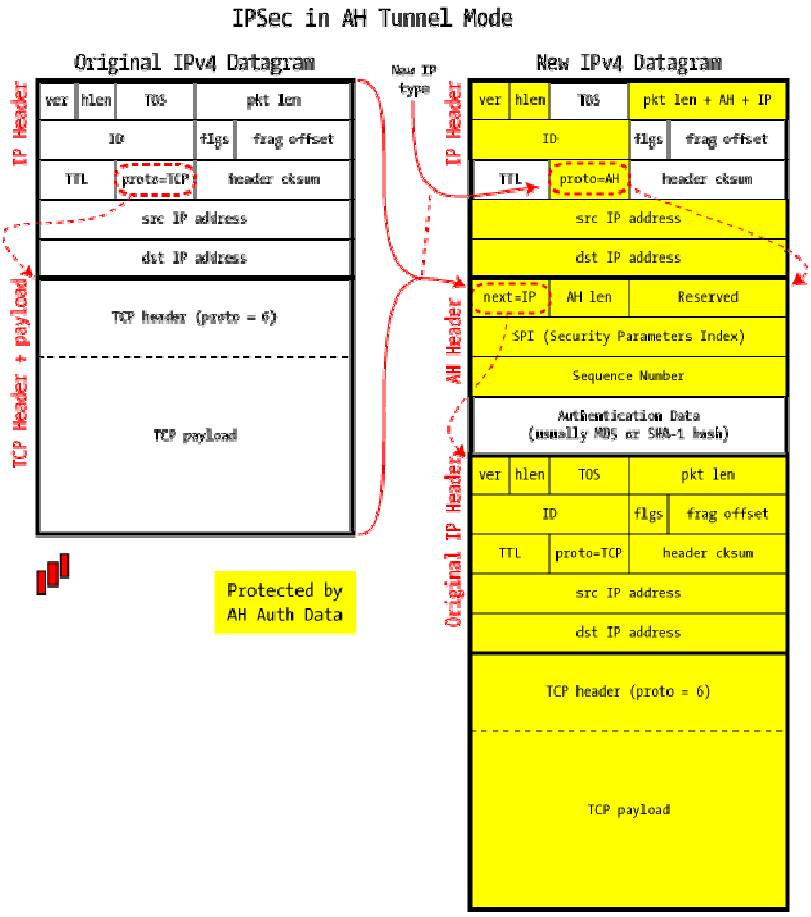


Figura 4 formato de protocolo IPSec para AH en modo Túnel

Diseñar e implantar la encriptación asimétrica y Autentificación de IPSEC

Algoritmos de autentificación

AH lleva un campo (Integrity Check Value) para comprobar la integridad del paquete y que nadie lo ha manipulado durante el trayecto. El valor de ese campo está dado por algoritmos de encriptación tales como MD5 o SHA-1.

Más que usar un checksum convencional, el cual podría no proveer una seguridad real contra una manipulación intencional, esta usa una Hashed Message Authentication Code (HMAC), que incorpora una clave secreta mientras se crea el hash. Aunque un atacante puede re calcular un hash fácilmente, sin la clave secreta no sería capaz de crear el ICV apropiado.

HMAC esta descrito por el RCF 2104 y se ilustra en la siguiente imagen:

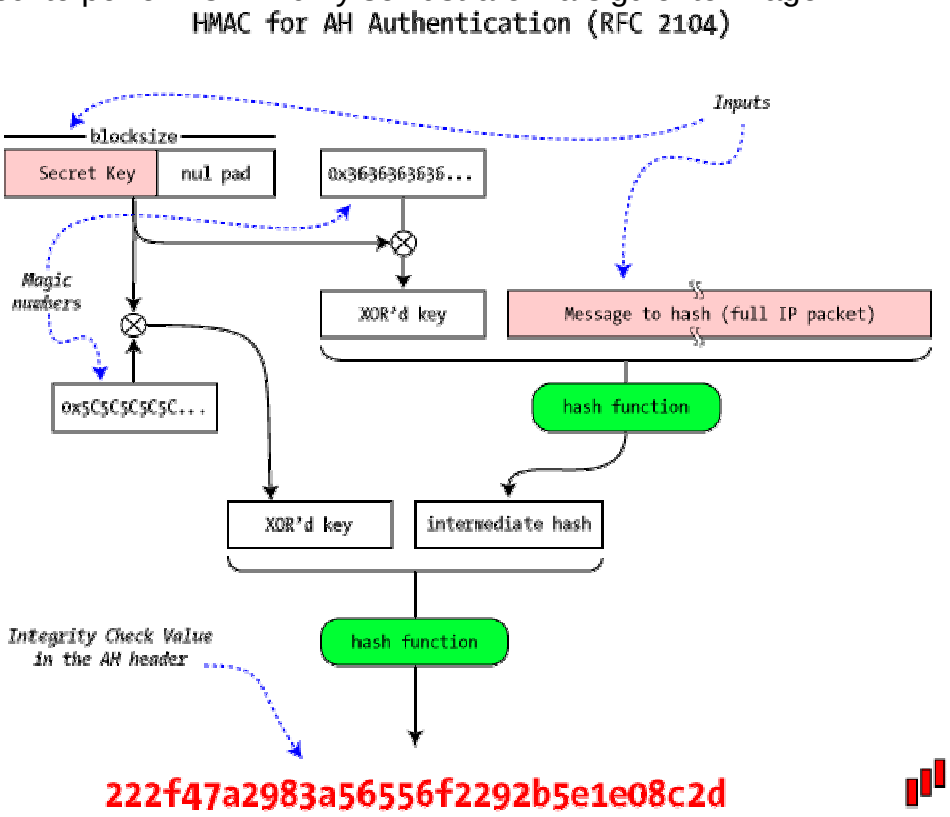


Figura 5: pseudocódigo de forma de autentificación

ESP (Encapsulating Security Payload)

Añadir encriptación hace que ESP sea un poco más complicado, ya que la encapsulación rodea a la carga útil es algo más que precederla con AH: ESP incluye cabecera y campos para dar soporte a la encriptación y a una autentificación opcional. Además, provee los modos de transporte y túnel, los cuales nos son ya familiares.

Las RFCs de IPsec no insisten demasiado en un sistema particular de encriptación, pero normalmente se utiliza DES, triple-DES, AES o Blowfish para asegurar la carga útil de “ojos indiscretos”. El algoritmo usado para una conexión en particular es definido por la Security Asociación (SA), y esta SA incluye no sólo la él algoritmo, también la llave usada.

A diferencia de AH, que da una pequeña cabecera antes de la carga útil, ESP rodea la carga útil con su protección. Los parámetros de seguridad Index y Sequence Number tienen el mismo propósito que en AH, pero nos encontramos como relleno en la cola del paquete el campo “siguiente campo” y el opcional “Authentication data”.

Es posible usar ESP sin ninguna encriptación (usar el algoritmo NULL), sin embargo estructura el paquete de la misma forma. No nos da ninguna confidencialidad a los datos que estamos transmitiendo, y sólo tiene sentido usarlo con una la autentificación ESP. No tiene sentido usar ESP sin encriptación o autentificación (a no ser que estemos simplemente probando el protocolo).

El relleno sirve para poder usar algoritmos de encriptación orientados a bloques, dado que tenemos que crear una carga a encripta que tenga un tamaño múltiplo de su tamaño de bloque. El tamaño del relleno viene dado por el campo pad len. El campo next hdr nos da el tipo (IP, TCP, UDP, etc) de la carga útil, aunque esto sea usado como un punto para volver hacia atrás en el paquete para ver que hay en el AH.

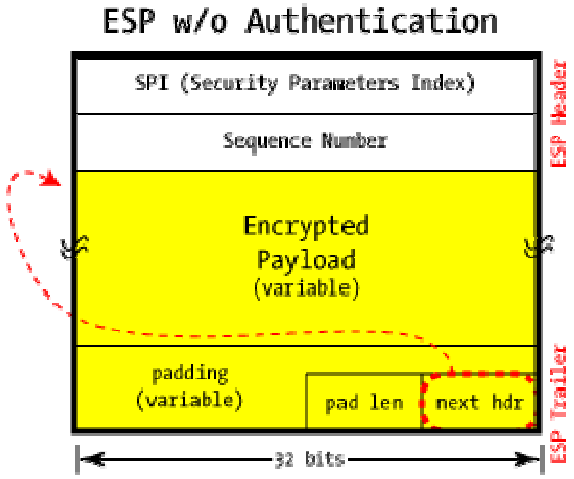
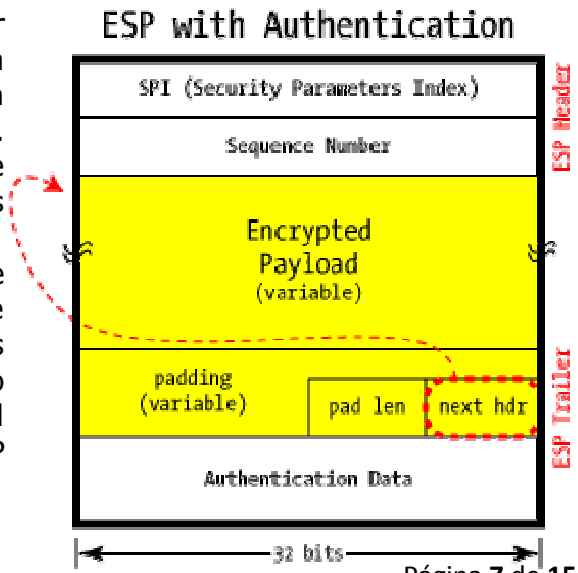


Figura 6 ESP W/O de Autentificación

El relleno sirve para poder usar algoritmos de encriptación orientados a bloques, dado que tenemos que crear una carga a encripta que tenga un tamaño múltiplo de su tamaño de bloque. El tamaño del relleno viene dado por el campo **pad len**. El campo **next hdr** nos da el tipo (IP, TCP, UDP, etc) de la carga útil, aunque esto sea usado como un punto para volver hacia atrás en el paquete para ver que hay en el AH.

Además de la encriptación, ESP puede proveer autentificación con la misma HMAC de AH. A diferencia de AH, esta **autentifica sólo la cabecera ESP y la carca útil encriptada**, no todo el paquete IP. Sorprendentemente, esto no hace que la seguridad de la autentificación más débil, pero nos da algunos beneficios importantes.

Cuando un forastero examina un paquete IP que contiene datos ESP, es prácticamente imposible adivinar que es lo que tiene dentro, excepto por los datos encontrados en la cabecera IP (siendo interesantes las direcciones IP de origen y destino). El atacante va a saber casi seguro que son datos ESP



(está en la cabecera que son datos ESP), pero no va a saber de qué tipo es la carga útil.

Incluso la presencia de Authentication Data no puede ser determina solamente con mirar al paquete. Esta resolución está hecha por la Security Parameters Index, que hace referencia al conjunto de parámetros pre compartidos para esta conexión.

6.1 ESP en Modo Túnel

El ESP en modo Túnel encapsula el datagrama IP entero y lo encripta:

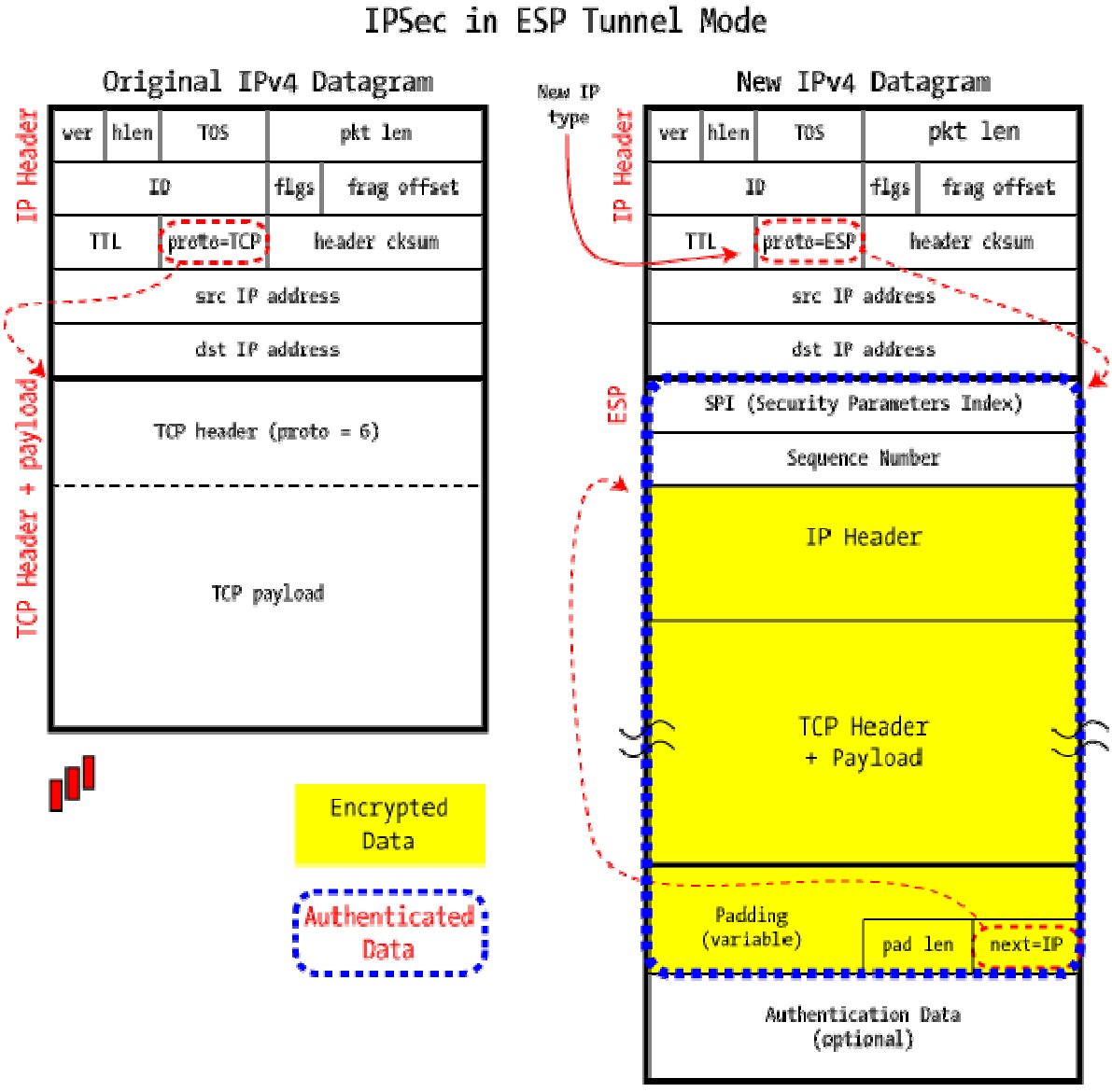


Figura 7 IPSec en modo tune para modo ESP

Proveer una conexión encriptada en modo túnel es dar una forma muy cercana a como se crea una VPN, y es lo que se nos viene a la cabeza a la mayoría cuando pensamos acerca de IPsec. Además de esto, tenemos que añadir autentificación. Esta parte se trata en la siguiente sección.
A diferencia de AH, donde un forastero puede ver fácilmente que es lo que se transmite en modo Túnel o Transporte, usando ESP eso no ocurre; esa información no está disponible para

el forastero. El caso es que en el modo túnel (poniendo **next=IP**), el datagrama IP entero original es parte de la carga útil encriptada, y no será visible para nadie que no pueda desencriptar el paquete.

Diseñar e implantar la programación en Dynamic para el micro controlador embebido Rabbit 3000.

En la siguiente imagen podemos ver parte de la programación en Dynamic C para programar los micros controladores, en especial vemos la parte de autentificación de usuario y contraseña para el acceso a nuestra red privada virtual y permitir el acceso a la página principal donde veremos la pagina de bienvenida mostrada mas tarde.

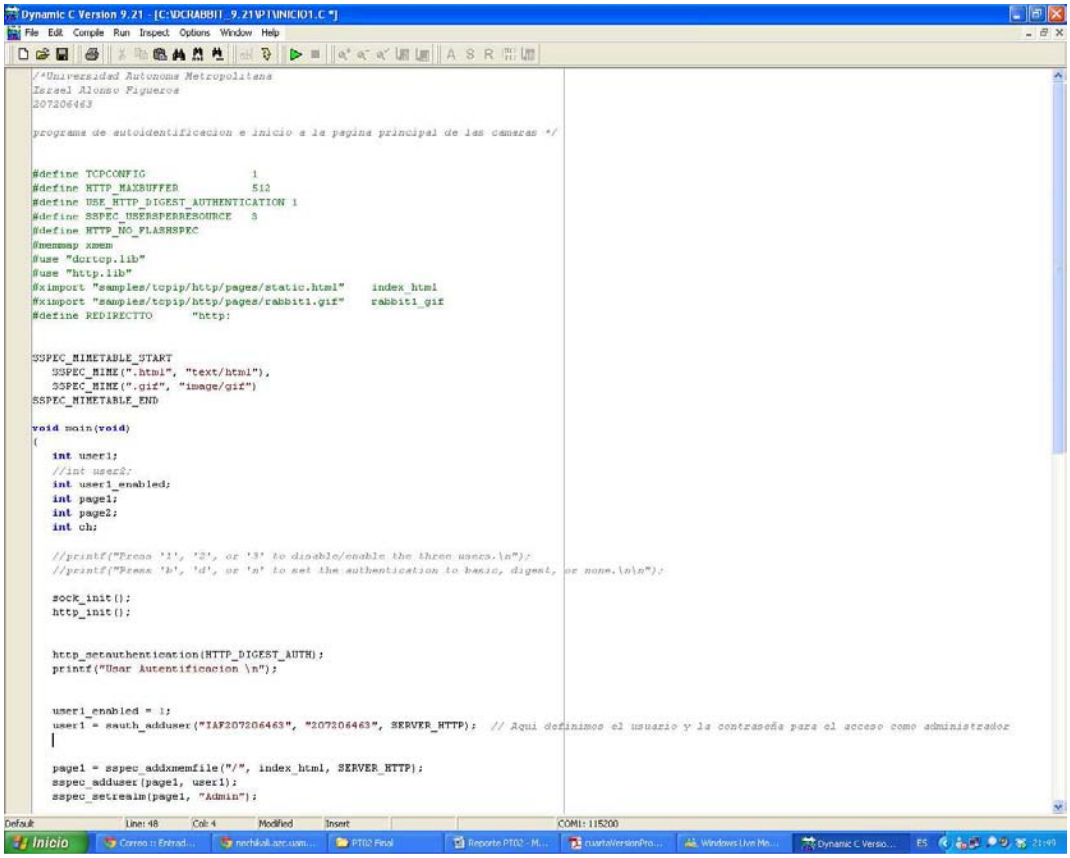


Figura 8 imagen de código para la programación en Dynamic C.

Implantación de bibliotecas de red para el micro controlador embebido Rabbit 3000.

Para la implementación de las bibliotecas de red para nuestro micro controladores Rabbit Editaremos como biblioteca principal la biblioteca tcp_config para asignar ip estática o dinámica dependiendo el uso, en nuestro caso lo realizamos como ip estática, como se muestra en la siguiente imagen.

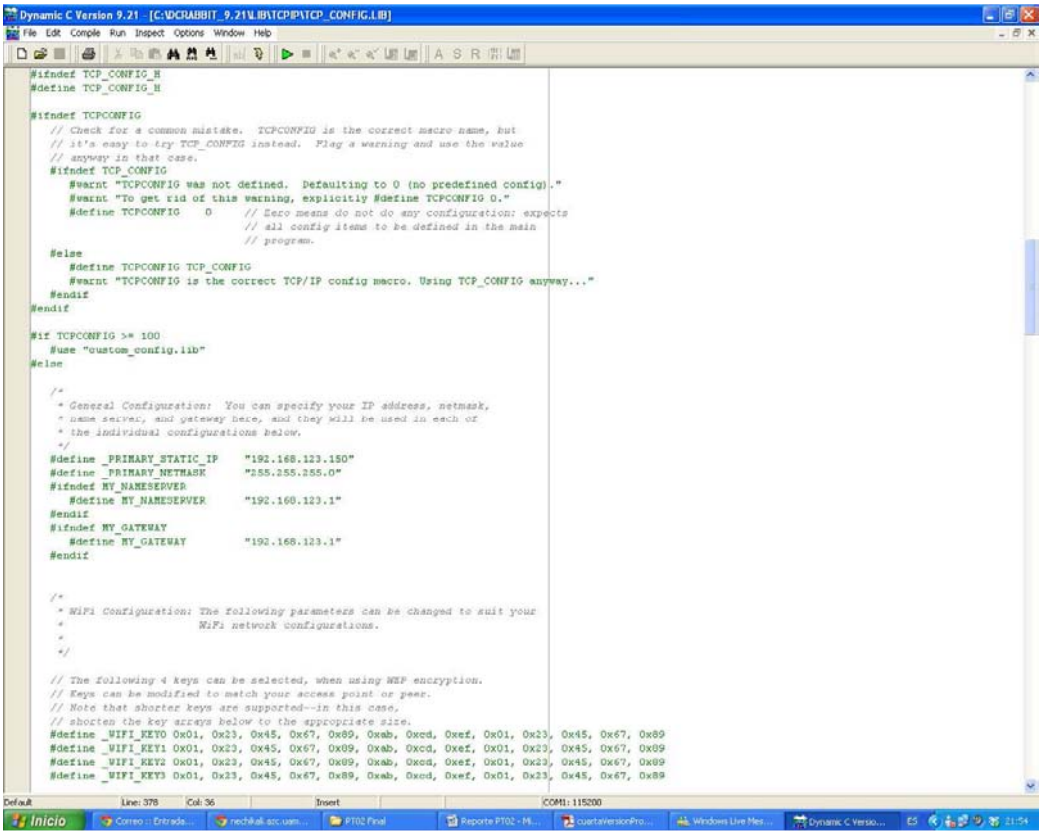


Figura 9: ilustración de programación de bibliotecas de red en Dynamic C.

Implantación de la comunicación física del micro controlador embebido Rabbit 3000 con las cámaras IP.

Para lograr la comunicación física de las cámaras IP con el micro controlador Rabbit 3000 hemos realizado la siguiente configuración de cable utp de Pc a Pc, como se muestra en la siguiente imagen.

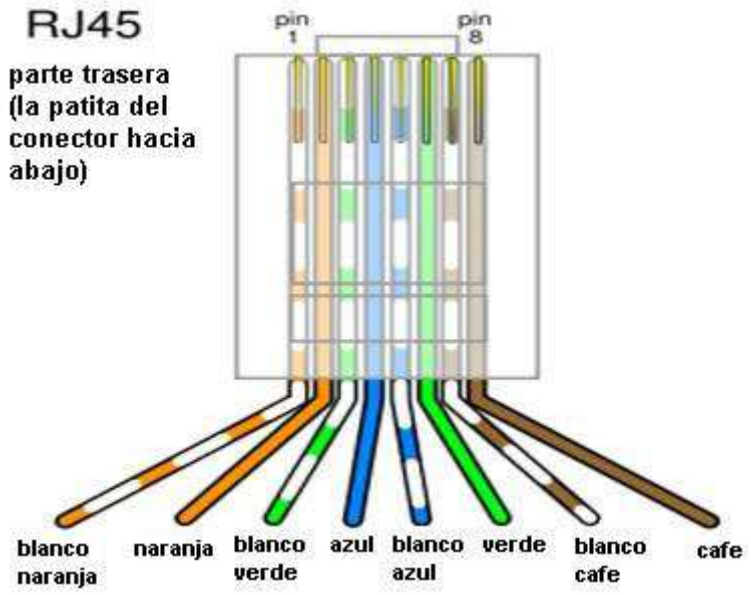


Figura 10: Configuración de cables de red

Y de este modo poder conectarlas con las cámaras AXIS 205 a través de uno Switch o concentrador hub para poder conectar todas las cámaras por medio de red Ethernet y tener acceso a ellas por medio de la implantación del punto siguiente.

Implantación de la comunicación física del micro controlador embebido Rabbit 3000 con el cliente que reside en el web browser.

Para lograr la comunicación física con uno del micro controlador Rabbit 3000 a través de su puerto de Ethernet 10/100 con el cliente que reside en el web browser lo que hacemos es proceder a la conexión del cable utp del tipo Pc a Pc (figura 1).

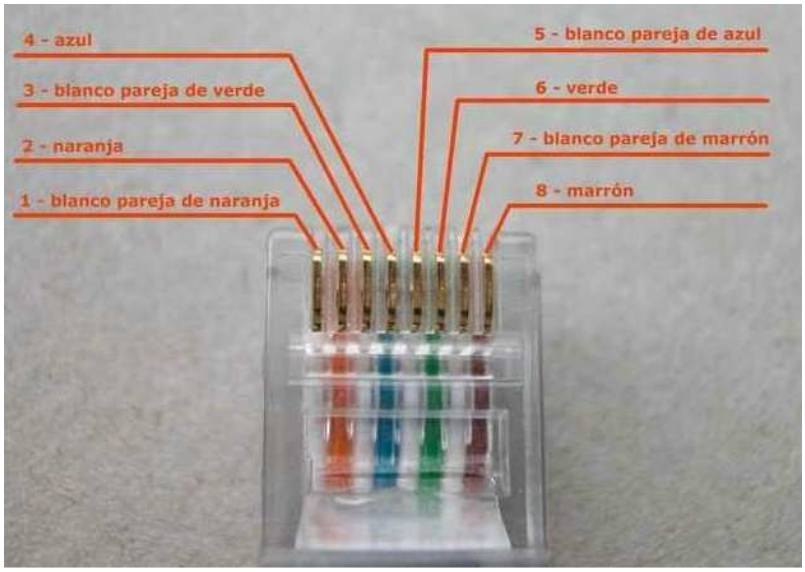


Figura 11: configuración de cable utp para conexión Pc a Pc

Figura 12 imagen que ilustra la conexión de micro controlador con cable UTP

Las dos últimas implementaciones son para poder hacer lo que se detalla en el siguiente diagrama de conexión y así poder realizar la programación de la Red Privada Virtual.

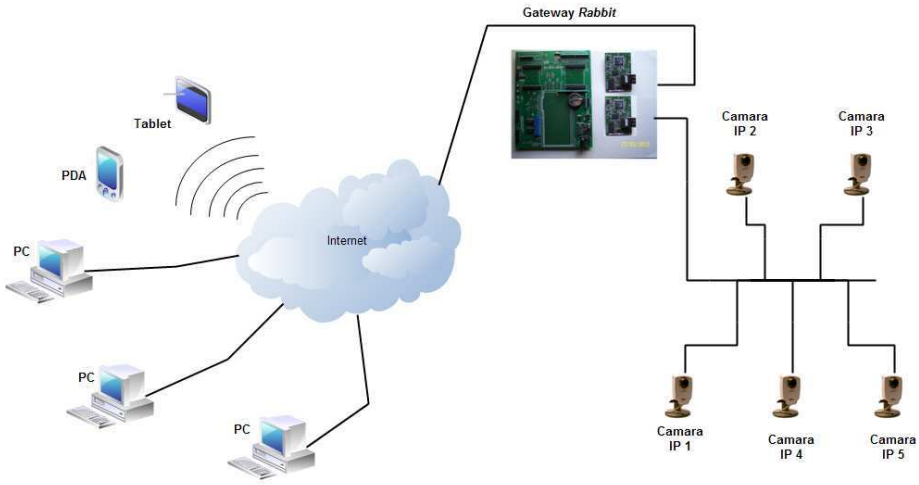


Figura 13 Diagrama ilustrativo de la conexión final

Configuración y conexión física

Para poder conectar la red Internet a nuestra red de cámaras es necesario agregar un segundo modulo micro controlador Rabbit como se muestra en la siguiente imagen.

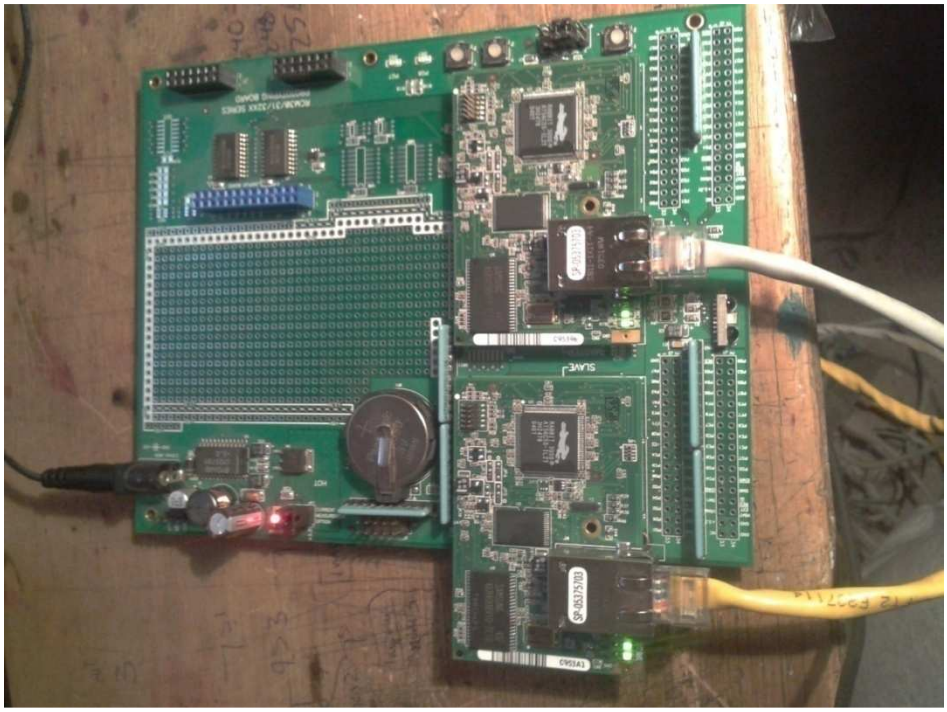


Figura 14: imagen de conexión física de los cables de red

Y de igual forma podemos ver el funcionamiento accediendo a nuestro web browser tecleando la ip 192.168.123.150 que esta misma es editable a través de la biblioteca *tcp_conf* agregada en el archivo de los entregables.

La implementación de Usuario y Contraseña para el acceso a nuestra modulo y así poder ver las cámaras tomamos en cuenta un algoritmo siguiente

Implantar el mecanismo de validación de usuario y contraseña

El proceso general de autenticación consta de los siguientes pasos:

1. El usuario solicita acceso a un sistema.
2. El sistema solicita al usuario que se autentique.
3. El usuario aporta las credenciales que le identifican y permiten verificar la autenticidad de la identificación.
4. El sistema valido según sus reglas si las credenciales aportadas son suficientes para dar acceso al usuario o no.

Con lo que nos muestra la siguiente imagen:

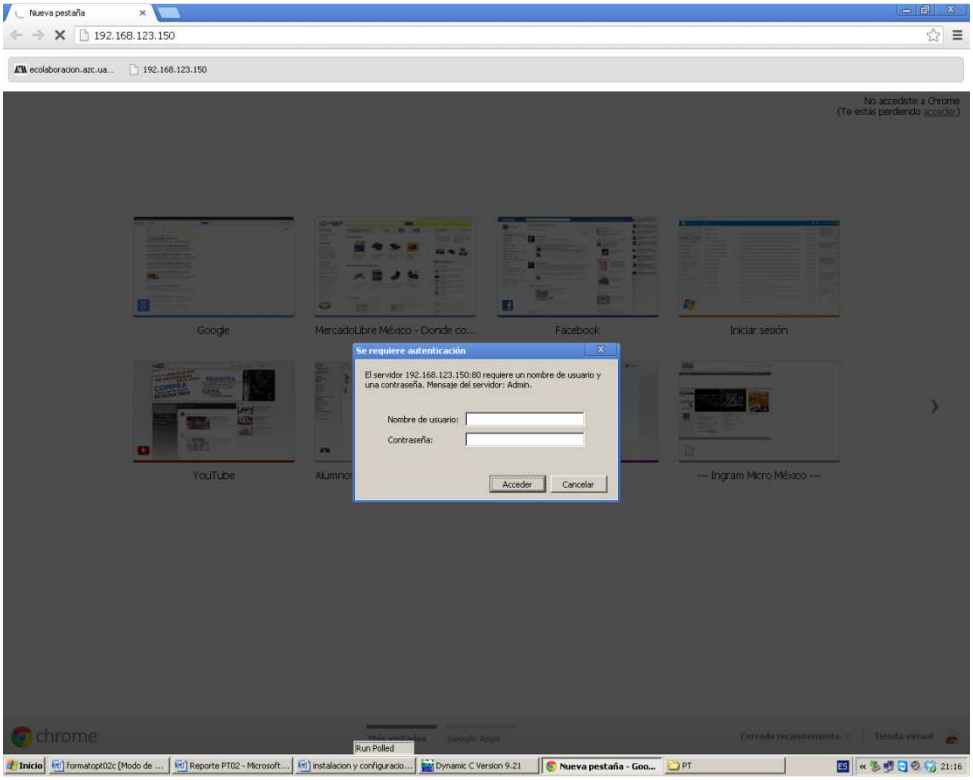


Figura 15: imagen de funcionamiento donde vemos la petición de autentificación

Donde nuestro usuario y contraseña los conoce solo el administrador del sistema y él es el único que puede dar de alta a un nuevo usuario. Para nuestro Administrador los campos son: Usuario: IAF207206463 y Contraseña: 207206463

Pruebas de funcionalidad de una red privada virtual implantada

En cuanto al plan de pruebas como primer paso es realizar la autentificación tecleando el usuario y contraseñas antes mencionadas para poder acceder a la página de inicio (index.html) hecho esto se nos mostrara la siguiente imagen:

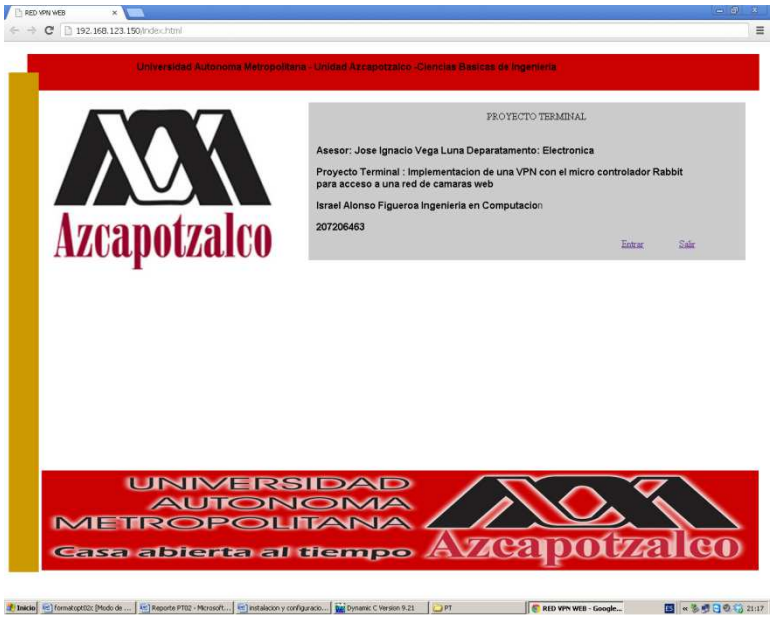


Figura 16: imagen de funcionamiento pantalla de bienvenida

Y para acceso a ver la red de cámaras web solo damos en el botón de entrar y nos mostrara la siguiente imagen donde podremos ver el acceso a las cámaras web y así seleccionar la cámara que queremos ver.

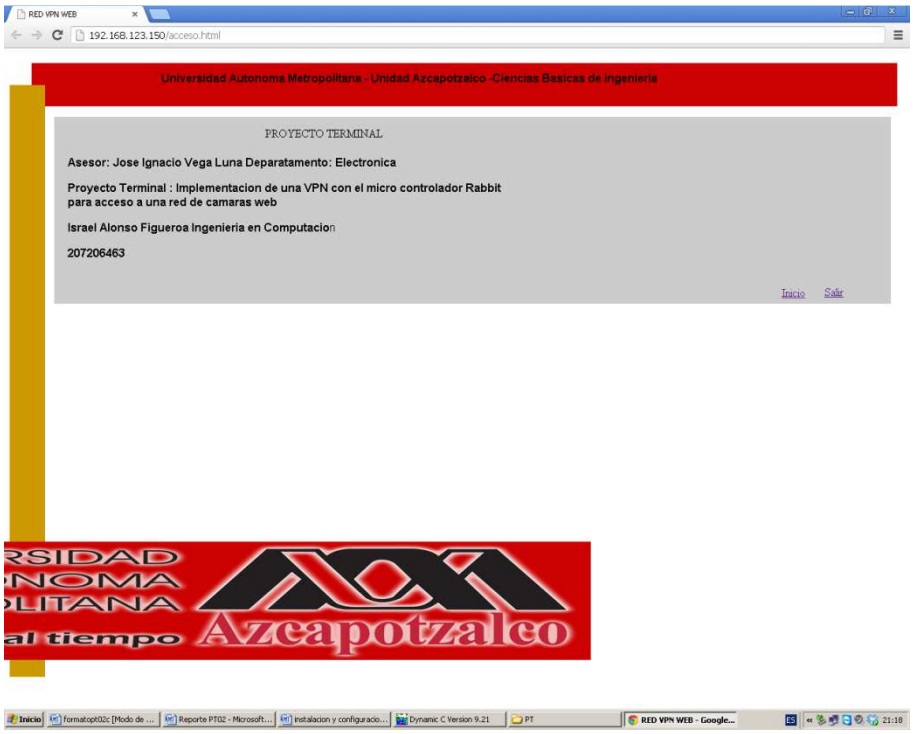


Figura 17: pantalla de funcionamiento

Conclusiones:

Durante la redacción de este documento a la aplicación se le estaban practicando más pruebas y una serie de revisiones para intentar detectar cualquier fallo que pudiese ser generado a causa de un defecto en alguno de los módulos del programa o, quizá, por defectos en la planeación del mismo. Hecho que favoreció en cuanto a la generación en tiempo real de información y por lo tanto, no caímos en la necesidad de elaborar una gran cantidad de versiones del reporte del proyecto terminal.

Bibliografía:

W. Scott et a., “Introduction to VPNs and IPSec Overview,” in IPsec VPN Design, First Printing, Indianapolis 46240 USA, April, 2005, pp. 3-19

T. M. Edison Rafael, “Diseño e implementación de una VPN en una empresa comercializadora utilizando IPSec” Proyecto previo a la obtención de Ingeniero en informática, Escuela politécnica Nacional, Ciudad de Quito, Marzo de 2006. Disponible en:

http://bibdigital.epn.edu.ec/bitstream/15000/214/1/CD-0210.pdf

C. C. Juan Alberto, “Diseño e implementación de una lan virtual con switches de red,” Proyecto Terminal de Ing. Computación, UAM-Azcapotzalco, Ciudad de México, D.F., México 2010. Disponible en:

http://espartaco.azc.uam.mx/tesis/X17194.pdf

P. L. Susana Elianeth, “Implementación servidores de acceso remoto de bajo costo en plataformas Windows y Linux,” Proyecto Terminal de Ing. Electrónica, Departamento de Electrónica, UAM-Azcapotzalco, Ciudad de México, D.F., México 2006. Disponible en:

http://espartaco.azc.uam.mx/tesis/x16590.pdf