# ¿Qué significado es MAC… **Qué es la dirección MAC de tu ordenador**

MAC es el acrónimo o abreviatura de **Media Access Control** y se refiere al conjunto de protocolos comunicacionales por medio del cual, varios equipos o dispositivos conectados a la red acuerdan usar conjuntamente un determinado medio de transmisión.

**La dirección MAC es un identificador único** que cada fabricante le asigna a la tarjeta de red de sus dispositivos conectados, desde un ordenador o móvil hasta routers, impresoras u otros dispositivos como tu Chromecast. Sus siglas vienen del inglés, y significan Media Access Control. Como hay dispositivos con diferentes tarjetas de red, como una para WiFi y otra para Ethernet, algunos pueden tener diferentes direcciones MAC dependiendo de por dónde se conecten.

Cuando estés revisando tu red para saber quién está conectado a ella, los programas que vas a utilizar te mostrarán determinada información sobre los dispositivos conectados. Por ejemplo, sabrás su fabricante, [su dirección IP](https://www.xatakamovil.com/conectividad/conoce-tu-router-i-direccion-ip-mascara-de-subred-y-puerta-de-enlace) y su MAC. Por lo general no cuesta demasiado identificar cada dispositivo, pero a veces tenemos tantos que la cosa se pone un poco complicada.

Por eso hoy te vamos a explicar **qué es la dirección MAC de un dispositivo** y cómo puedes saber la del tuyo. De esta manera, una vez identificada la de tu móvil u ordenador te será mucho más fácil identificarlo en la lista de dispositivos conectados.

## ¿Qué es el MAC de un dispositivo?

**La dirección MAC es un identificador único** que cada fabricante le asigna a la tarjeta de red de sus dispositivos conectados, desde un ordenador o móvil hasta routers, impresoras u otros dispositivos como tu Chromecast. Sus siglas vienen del inglés, y significan Media Access Control. Como hay dispositivos con diferentes tarjetas de red, como una para WiFi y otra para Ethernet, algunos pueden tener diferentes direcciones MAC dependiendo de por dónde se conecten.

Las direcciones MAC están formadas por 48 bits representados generalmente por [dígitos hexadecimales](https://es.wikipedia.org/wiki/Sistema_hexadecimal). Como cada hexadecimal equivale a cuatro binarios (48:4=12), la dirección acaba siendo formada por **12 dígitos agrupados en seis parejas** separadas generalmente por dos puntos, aunque también puede haber un guión o nada en absoluto. De esta manera, un ejemplo de dirección MAC podría ser 00:1e:c2:9e:28:6b.

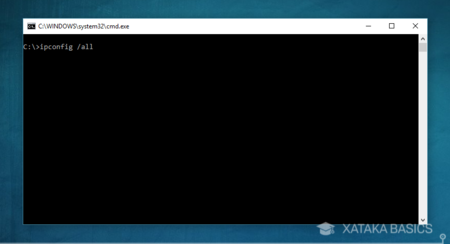
Otra cosa que tienes que tener en cuenta es que la mitad de los bits de una dirección MAC, tres de las seis parejas, identifican al fabricante, y la otra mitad al modelo. Por ejemplo, los números 00:1e:c2 del ejemplo de dirección **pertenecen siempre al fabricante Apple Inc**, mientras que los últimos seis determinan el modelo de dispositivo. Hay [buscadores especializados](http://www.coffer.com/mac_find/?string=00:1E:C2) para saber el fabricante de un dispositivo dependiendo de los primeros seis dígitos de su MAC.

Como son identificadores únicos, las MAC pueden ser utilizadas por un administrador de red para permitir o denegar el acceso de determinados dispositivos a una red. En teoría son fijas para cada dispositivo, aunque **existen maneras de cambiarlas** en el caso de que quieras hacerlas más reconocibles en tu red o evitar bloqueos.

Esta exclusividad de cada MAC hacia un único dispositivo también exige que tengas especial cuidado. Por ejemplo, cuando te conectas o intentas conectarte a un router, **tu móvil u ordenador le enviará automáticamente su MAC**. Es una de las razones por las que tienes que saber siempre dónde te conectas a Internet y a quién le pertenece esta red.

## ¿Cómo puedo obtener el MAC de mi dispositivo?

Si estás revisando quién está conectado a tu red y quieres identificar cada dispositivo tienes dos maneras de hacerlo. Puedes ir encendiendo y apagando cada dispositivo, pero también puedes ir a tu móvil u ordenador, **averiguar su MAC** y buscarla en la lista de conectados. Te vamos a decir cómo hacerlo con cada sistema operativo.Obtener el MAC en Windows



* Pulsa las teclas Windows + R para abrir Ejecutar.
* Escribe cmd y presiona Enter para ir al Símbolo de sistema
* Escribe ipconfig /all.
* En la entrada Dirección física te dirá la dirección MAC de tu ordenador.

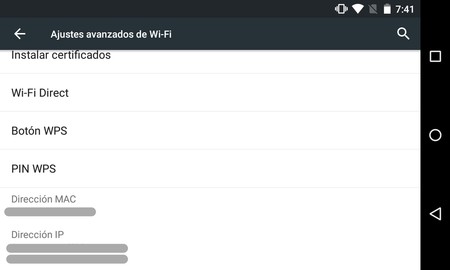
#### Obtener el MAC en macOS

* Abre las Preferencias del Sistema.
* Haz click en Red, y en el panel de la izquierda selecciona en la que estás conectado.
* Pulsa sobre el botón Avanzado en la parte inferior de la ventana.
* Elige la pestaña Hardware, y en ella tienes tu MAC.

#### Obtener el MAC en GNU/Linux

* Ve a la consola del sistema.
* Escribe ifconfig.
* El MAC es la dirección del campo HWaddr.

#### Obtener el MAC en Android



* Entra en Configuración.
* Pulsa sobre Wi-Fi, y elige la opción Ajustes avanzados.
* La dirección MAC te aparecerá al final de la pantalla.

#### Obtener el MAC en iOS

* Entra en Ajustes.
* Pulsa sobre General, y dentro elige la opción Información.
* En el campo Dirección Wi-Fi tendrás tu MAC.

<https://www.youtube.com/watch?v=QIU62diZgNI>

# **C# / ¿Cómo generar y usar licencias para activar tus proyectos? –**

# **Respondiendo Preguntas #1**

CÓDIGO PARA OBTENER DIRECCIÓN MAC:

public string GetMacAddress()

{ NetworkInterface[] nics = NetworkInterface.GetAllNetworkInterfaces();

String sMacAddress = string.Empty;

foreach (NetworkInterface adapter in nics) {

if (sMacAddress == String.Empty)

{ IPInterfaceProperties properties = adapter.GetIPProperties(); sMacAddress = adapter.GetPhysicalAddress().ToString(); } }

return sMacAddress }

**Respuesta: Como crear licencias para un software realizado en JAVA**

Para crearle una licencia a un producto tienes primero que obtener identificadores únicos de la maquina. Estos te lo pueden dar la MAC, el id de la placa base y tambien puedes buscar en componentes como el MIcro y disco duro. Ahora hay varias formas de acceder a estas propiedades y es con el uso de procesos que ejecutan consultas a las clases del win32\_clase del sistema. Estas query las puedes probar en el sofware de windows Wbemtest. Algunas clases que te dan datos importantes de la maquina son:  
Win32\_BaseBoard  
Win32\_DiskDrive  
Win32\_Processor

También puedes obtener los datos de una forma mas facil, con la libreria jacob-1.17-M2  
Ejemplo:  
  
ActiveXComponent a= new ActiveXComponent("winmgmts:\\\\.");  
Variant b= a.invoke("InstancesOf", "Win32\_BaseBoard"); // para la placa base  
Enumeration<Variant> baseBoard = new EnumVariant(b.getDispatch());  
  
while (baseBoard .hasMoreElements())  
{  
ActiveXComponent cc= new ActiveXComponent(baseBoard .nextElement().getDispatch());  
  
System.out.println("Id de la placa base" + cc.getPropertyAsString("SerialNumber"));  
break;  
}  
  
//tienes que agregar la libreria y poner este metodo en el main

# **C# / ¿Cómo generar y usar licencias para activar tus proyectos? - Respondiendo Preguntas #1**

<https://github.com/RJCodeAdvance/Dasnet/blob/main/RepositorioMaestro.cs>

https://github.com/RJCodeAdvance/Dasnet/blob/main/Repositorio.cs

|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Data; |
|  | using System.Data.SqlClient; |
|  | using System.Linq; |
|  | using System.Text; |
|  | using System.Threading.Tasks; |
|  |  |
|  | namespace SoftwareDasnet.CapaDatos |
|  | { |
|  | public abstract class RepositorioMaestro:Repositorio |
|  | { |
|  | protected List<SqlParameter> parametros; |
|  | protected void ExecuteNonQuery(string transactSql) //Ejecutar sentencias de texto insert, update, delete con parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.Text; |
|  | foreach (SqlParameter item in parametros) |
|  | { |
|  | comando.Parameters.Add(item); |
|  | } |
|  | comando.ExecuteNonQuery(); |
|  | parametros.Clear(); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | protected void ExecuteNonQuery(string transactSql) //Ejecutar sentencias de texto insert, update, delete etc sin parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.Text; |
|  | comando.ExecuteNonQuery(); |
|  | parametros.Clear(); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | protected void ExecuteNonQuery(string transactSql) //Ejecutar procedimientos almacenados sin parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.StoredProcedure; |
|  | comando.ExecuteNonQuery(); |
|  | parametros.Clear(); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | protected void ExecuteNonQuery(string transactSql) //Ejecutar procedimientos almacenados con parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.StoredProcedure; |
|  | foreach (SqlParameter item in parametros) |
|  | { |
|  | comando.Parameters.Add(item); |
|  | } |
|  | comando.ExecuteNonQuery(); |
|  | parametros.Clear(); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | protected DataTable ExecuteReader(string transactSql) //Devolver tablas ejecutando consultas de texto con parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.Text; |
|  | foreach (SqlParameter item in parametros) |
|  | { |
|  | comando.Parameters.Add(item); |
|  | } |
|  | SqlDataReader lector = comando.ExecuteReader(); |
|  | using (var tabla = new DataTable()) |
|  | { |
|  | tabla.Load(lector); |
|  | lector.Dispose(); |
|  | return tabla; |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | protected DataTable ExecuteReader(string transactSql) //Devolver tablas ejecutando consultas de texto sin parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.Text; |
|  | SqlDataReader lector = comando.ExecuteReader(); |
|  | using (var tabla = new DataTable()) |
|  | { |
|  | tabla.Load(lector); |
|  | lector.Dispose(); |
|  | return tabla; |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | protected DataTable ExecuteReader(string transactSql) //Devolver tablas ejecutando consultas de procedimientos almacenados con parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.StoredProcedures; |
|  | foreach (SqlParameter item in parametros) |
|  | { |
|  | comando.Parameters.Add(item); |
|  | } |
|  | SqlDataReader lector = comando.ExecuteReader(); |
|  | using (var tabla = new DataTable()) |
|  | { |
|  | tabla.Load(lector); |
|  | lector.Dispose(); |
|  | return tabla; |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | protected DataTable ExecuteReader(string transactSql) //Devolver tablas ejecutando procedimientos almacenados sin parametros |
|  | { |
|  | using (var conexion = ObtenerConexion()) |
|  | { |
|  | conexion.Open(); |
|  | using (var comando = new SqlCommand()) |
|  | { |
|  | comando.Connection = conexion; |
|  | comando.CommandText = transactSql; |
|  | comando.CommandType = CommandType.StoredProcedure; |
|  | SqlDataReader lector = comando.ExecuteReader(); |
|  | using (var tabla = new DataTable()) |
|  | { |
|  | tabla.Load(lector); |
|  | lector.Dispose(); |
|  | return tabla; |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |

**Footer**

© 2023 GitHub, Inc.

**Footer navigation**

* [Terms](https://docs.github.com/en/github/site-policy/github-terms-of-service)
* [Privacy](https://docs.github.com/site-policy/privacy-policies/github-privacy-statement)
* [Security](https://github.com/security)
* [Status](https://www.githubstatus.com/)
* [Docs](https://docs.github.com/)
* [Contact GitHub](https://support.github.com/?tags=dotcom-footer)

|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Data.SqlClient; |
|  | using System.Linq; |
|  | using System.Text; |
|  | using System.Threading.Tasks; |
|  |  |
|  | namespace SoftwareDasnet.CapaDatos |
|  | { |
|  | public abstract class Repositorio |
|  | { |
|  | private readonly string CadenaConexion; |
|  | public Repositorio() |
|  | { |
|  | CadenaConexion = "server = TuServer; database=TuDataBase; integrated security=true"; |
|  | } |
|  | protected SqlConnection ObtenerConexion() |
|  | { |
|  | return new SqlConnection(CadenaConexion); |
|  | } |
|  | } |
|  | } |

PERIODO DE TIEMPO: En la base de datos declaramos un campo para la fecha de vigencia, pero como mencioné hay muchas formas de tratarlo, puedes cambiar el estado de activa a vencida, y hacer que el usuario se vuelva a comunicar contigo para adquirir una nueva licencia o renovar la que se tiene. Si tienes más dudas sobre el tema escríbenos en los comentarios

<https://foxlearn.com/articles/create-license-key-in-csharp-part1-210.html>

# JAVA **Licencias en una aplicación - Java**

<https://www.youtube.com/watch?v=zWesRynpWlg>

[Dasnet](https://www.youtube.com/playlist?list=PLqjdFmR_HdQQHWmVVik6ohzoNhi67CBEo)

# **C# / ¿Cómo generar y usar licencias para activar tus proyectos? - Respondiendo Preguntas #1**

<https://www.youtube.com/watch?v=QIU62diZgNI>