



Los alumnos deben llenar esta hoja y entregarla al supervisor junto con la versión final de su monografía.

Número de convocatoria del alumno

Nombre y apellido(s) del alumno

Número del colegio

Nombre del colegio

Convocatoria de exámenes (mayo o noviembre)

MAYO

Año

2013

Asignatura del Programa del Diploma en la que se ha inscrito la monografía: INFORMÁTICA

(En el caso de una monografía en lenguas, señale si se trata del Grupo 1 o el Grupo 2.)

Título de la monografía: PUEDA VISUAL BASIC AYUDAR A GESTIONAR UNA  
ENBARCACION MARITIMA?

### Declaración del alumno

*El alumno debe firmar esta declaración; de lo contrario, es posible que no reciba una calificación final.*

Confirmando que soy el autor de este trabajo y que no he recibido más ayuda que la permitida por el Bachillerato Internacional.

He citado debidamente las palabras, ideas o gráficos de otra persona, se hayan expresado estos de forma escrita, oral o visual.

Sé que el máximo de palabras permitido para las monografías es 4.000, y que a los examinadores no se les pide que lean monografías que superen ese límite.

Esta es la versión final de mi monografía.

Firma del alumno:

Fecha:

## Informe y declaración del supervisor

El supervisor debe completar este informe, firmar la declaración y luego entregar esta portada junto con la versión final de la monografía al coordinador del Programa del Diploma.

Nombre y apellido(s) del supervisor [MAYÚSCULAS]:

Si lo considera adecuado, escriba algunos comentarios sobre el contexto en que el alumno desarrolló la investigación, las dificultades que encontró y cómo las ha superado (ver página 13 de la guía para la monografía). La entrevista final con el alumno puede ofrecer información útil. Estos comentarios pueden ayudar al examinador a conceder un nivel de logro para el criterio K (valoración global). No escriba comentarios sobre circunstancias adversas personales que puedan haber afectado al alumno. En el caso en que el número de horas dedicadas a la discusión de la monografía con el alumno sea cero, debe explicarse este hecho indicando cómo se ha podido garantizar la autoría original del alumno. Puede adjuntar una hoja adicional si necesita más espacio para escribir sus comentarios.

EL DESARROLLO DE LA MONOGRAFÍA HA SIDO ESPECIALMENTE DESTACADO EN EL CASO DE [REDACTED] YA QUE HA IMPLICADO UN ESFUERZO NOTABLE EN LA AUTOFORMACIÓN EN EL LENGUAJE DE PROGRAMACIÓN USADO. CABE DESTACAR QUE SU CONOCIMIENTO INICIAL ERA PRÁCTICAMENTE NULO Y QUE EL OBJETIVO PLANTADO INICIALMENTE ERA LA CREACIÓN DE UNA HERRAMIENTA ÚTIL Y NUEVA, NO EL COMPENDIO DE FUNCIONES PREESTABLECIDAS.

EN EL PROCESO DE BÚSQUEDA Y INVESTIGACIÓN HA SIDO ABSOLUTAMENTE AUTÓNOMO, TENIENDO LA CAPACIDAD DE READAPTAR EL ENFOQUE CUANDO COMPROBÓ QUE LOS APARTADOS DE NAVEGACIÓN Y METEOROLOGÍA IMPLICABAN MENOS ESFUERZO DEL ESPERADO. EN ESE MOMENTO SE PLANTEÓ EL AMPLIAR EL ABANICO A LA CONSTRUCCIÓN DEL ELEMENTO EMISOR DE CÓDIGO MORSE (ALTAZOR MODIFICADO).

SU CAPACIDAD DE ADAPTACIÓN HA SIDO MUY DESTACABLE.

El supervisor debe firmar esta declaración; de lo contrario, es posible que no se otorgue una calificación final.

He leído la versión final de la monografía, la cual será entregada al examinador.

A mi leal saber y entender, la monografía es el trabajo auténtico del alumno.

He dedicado  horas a discutir con el alumno su progreso en la realización de la monografía.

Firma del supervisor:

Fecha:

# Formulario de evaluación (para uso exclusivo del examinador)

Criterios de evaluación	Nivel de logro					
	Examinador 1	Máximo	Examinador 2	Máximo	Examinador 3	
A Formulación del problema de investigación	2	2		2		
B Introducción	1	2		2		
C Investigación	3	4		4		
D Conocimiento y comprensión del tema	4	4		4		
E Argumento razonado	4	4		4		
F Aplicación de habilidades de análisis y evaluación apropiadas para la asignatura	3	4		4		
G Uso de un lenguaje apropiado para la asignatura	3	4		4		
H Conclusión	2	2		2		
I Presentación formal	4	4		4		
J Resumen	2	2		2		
K Valoración global	4	4		4		
Total (máximo 36)	32					

**Puede Visual  
Basic ayudar a  
gestionar una  
embarcación  
marítima?**

Grupo: 2 Bachillerato A

Tutoría:

Núm. Palabras: 3994

---

## Resumen

El objetivo de INA, el Programa de Navegación Marítima, es hacer una herramienta que pueda servir de ayuda a embarcaciones con acceso a Internet, facilitándoles informaciones esenciales como son la meteorología, la posición en un GPS y ayudarles con la comunicación. Además, al ser principiantes en el mundo de la programación informática, pretendemos familiarizarnos y adquirir conocimientos básicos sobre este mundo, para futuros proyectos.

El programa cuenta de tres apartados: un navegador que nos dirigirá a las páginas web que contengan informaciones meteorológicas; un enlace a un programa GPS que nos mostrará nuestra posición en Google Maps y nos permitirá compartirla con otros usuarios del programa; y un traductor morse, que permitirá aumentar la velocidad de emisión y transmisión de mensajes a través de este lenguaje, que esta cayendo en desuso a pesar de su utilidad. Para exteriorizar el sonido hemos construido un soporte hardware que funciona como amplificador.

Hemos podido comprobar, a partir de la formación que requería el nivel del programa logrado, que el Visual Basic es la clase de lenguaje adecuado para iniciarse en la programación, pero, cuando los proyectos deseados sean relativamente ambiciosos, es un tanto limitado y dependerá de complementos y librerías varias.

INA, como programa, presenta algunas limitaciones deudas al lenguaje con el que ha sido programado, pero aún así ha cumplido los objetivos satisfactoriamente, y se presenta como una herramienta bastante útil.

Quiero agradecer la colaboración de [redacted] sin sus ironías constantes e inestimable ayuda este trabajo no hubiera sido posible; al tutor, [redacted], por sus grandes consejos y por estar ahí en los momentos complicados; a [redacted] para orientarme en el día a día y animarme cuando era necesario; a todo el equipo de OPTIMUS, por facilitarme el programa y aconsejarme; pero por encima de todos, este trabajo lo tengo que agradecer a mi abuelo, [redacted], sin él, su perseverancia, las clases conjuntas de programación, su seguimiento constante, por estar ahí en todo momento, su experiencia, su paciencia... Por todo esto y mucho más, gracias.

# Índice de Contenidos

<b>1. <u>Introducción al Programa de navegación marítima</u></b> .....	<b>4</b>
<b>2. <u>Objetivos</u></b> .....	<b>5</b>
<b>3. <u>Antes de programar</u></b> .....	<b>6</b>
<b>4. <u>INA, El Programa de Navegación Marítima</u></b> .....	<b>7</b>
4.1. <u>Pantalla Menú</u> .....	7
4.2. <u>Meteorología</u> .....	9
4.2.1. <u>Objetivo</u> .....	9
4.2.2. <u>Descripción Funciones</u> .....	9
4.2.3. <u>Explicación del código</u> .....	12
4.2.4. <u>Conclusiones</u> .....	15
4.3. <u>Localización</u> .....	15
4.3.1. <u>Objetivo</u> .....	15
4.3.2. <u>Descripción del programa GPS</u> .....	15
4.3.3. <u>Explicación del código</u> .....	18
4.3.4. <u>Conclusiones</u> .....	18
4.4. <u>Comunicación</u> .....	19
4.4.1. <u>Objetivos</u> .....	19
4.4.2. <u>Introducción al Código Morse</u> .....	19
4.4.3. <u>Descripción de funciones</u> .....	21
4.4.4. <u>Explicación del código</u> .....	21
4.4.5. <u>Construcción del altavoz</u> .....	27
4.4.6. <u>Conclusiones</u> .....	27
<b>5. <u>Conclusiones generales</u></b> .....	<b>28</b>
<b>6. <u>Bibliografía</u></b> .....	<b>30</b>

---

# 1. Introducción al Programa de navegación marítima

Este trabajo está dirigido a todos los amantes del mundo marítimo que poseen una embarcación con acceso a Internet y necesiten un programa de navegación que, además de mostrarles su posicionamiento en coordenadas, les pueda informar sobre meteorología y ayudarles con la comunicación con otras embarcaciones.

Mi afición al mar y a los barcos proviene de familia, ya desde pequeño mi abuelo me inició a la navegación marítima llevándome de paseo con su velero (nombrado INA, de aquí el título del trabajo). Desde entonces he ido aprendiendo como se debe manejar el timón, a arriar las velas, a leer y a utilizar cartas náuticas, etc. Pero sobretodo, he aprendido a respetar el mar; los días de tempestad cerrado dentro del camarote, me hicieron dar cuenta de los peligros del mundo marítimo y de que si necesitaba ayuda de otros barcos me podía comunicar con ellos a través de códigos como el morse.

Este tipo de código, a pesar de ser muy conocido, es difícil de aprender y se está perdiendo, por lo tanto no todo el mundo lo puede hacer funcionar, pero sí que todo el mundo puede tener problemas; por lo tanto, decidí hacer este trabajo para facilitar la comunicación entre barcos y facilitarles otros tipos de servicios, como el meteorológico y el de posicionamiento. Además, con este trabajo, puedo dar a conocer, o mejor dicho, potenciar un lenguaje que se está perdiendo a pesar de su utilidad.

## 2. Objetivos

Con este trabajo, aparte de introducirnos en el mundo de la programación informática, queremos comprobar que la programación perteneciente a Visual Basic nos puede ayudar a hacer un programa de calidad relativamente elevada que sea útil para los navegantes, a la vez que aprendemos a programar con un lenguaje bastante corriente como es el Basic, lo que nos puede servir en proyectos futuros.



---

El objetivo principal es construir una herramienta de software que pueda facilitar todo tipo de tareas a las embarcaciones marítimas que usen un ordenador portátil con acceso a Internet. Por esta razón deberemos incluir dentro del programa de navegación aplicaciones para navegar más fácilmente como el acceso a información referente a la meteorología y al posicionamiento y la facilitación de la comunicación.

### 3. Antes de programar

Hacer una herramienta que sirviera para facilitar la vida a los navegantes poseedores de un portátil con acceso a Internet requería conocimientos de programación, que en mi caso no eran demasiados. Al tenerme que formar para realizar el proyecto, decidí consultar los distintos lenguajes existentes con los que podría trabajar y hacer un programa como es debido que pudiera ser realmente útil para conseguir información imprescindible para la vida en alta mar, a la vez que no me resultara una tarea ardua y me costara mucho tiempo aprender a codificarlo. Por esta razón me decidí por el programa *Microsoft Visual Basic 2010 Express Edition* y el lenguaje *Visual Basic.NET*.

Como el nombre indica, el *Visual Basic* es un lenguaje bastante intuitivo y, para un principiante como yo, el aprendizaje es notablemente más rápido que con cualquier otro lenguaje. Además, al ser un de los lenguajes de uso más extendido, resulta fácil encontrar información, documentación y fuentes para los proyectos, además de ser fácilmente extensible mediante librerías DLL y componentes ActiveX de otros lenguajes.

El programa que ha servido para realizar este proyecto fue facilitado por la empresa de comunicaciones *OPTIMUS*, con todos los derechos de autor y privilegios.

---

4.

INA

## El Programa de Navegación Marítima

Este apartado va a tratar sobre el software del proyecto: se va a comentar cada una de las funciones del código utilizado para conseguir la depuración deseada, las funciones de cada uno de los apartados y la forma en que funcionan y, finalmente, se comentarán y analizarán los problemas potenciales que han surgido a lo largo del proceso de programación.

Las partes comentadas corresponderán con las distintas partes del programa y se presentarán siguiendo el siguiente esquema:

- Objetivos de cada apartado
- Descripción de sus distintas funciones
- Análisis y explicación del texto de programación
- Conclusión (problemas, comentarios y sensaciones)

### PANTALLA MENÚ

Esta es la ventana inicial del programa. El diseño que se puede apreciar debajo es propio, y ha sido confeccionado con el *Paint* a partir de una imagen prediseñada de Internet.



---

El objetivo principal de la ventana inicial es llamar la atención, ser atractiva a la vista y captivar al público con su diseño.

En ella sólo se han utilizado cuatro botones:

- cdmposició (posicionamiento)
- cdmcomunicació (comunicación)
- cdmmeteorologia (meteorología)
- Sortir

Cada uno de estos botones, excepto el de salir (Sortir), lleva a su correspondiente apartado del programa. Esta acción la realizan gracias a las funciones .Show, que nos abre la destinación que le indiquemos (en este caso el formulario Form5), y .Hide, que nos cierra la ventana indicada (en este caso, la que nos encontramos).

```
Private Sub cdmposicio_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cdmposicio.Click
    Form5.Show()
    Me.Hide()
    Shell("C:\Program Files (x86)\gps\pcgps.exe")
End Sub
```

Este ejemplo es el de posicionamiento, que además de abrirnos su respectivo formulario, nos abre el programa (*Google Maps GPS Tracker*) que aparece en la dirección descrita dentro de la función Shell.

La complicación más importante que tiene la ventana inicial es encontrar las funciones claves que te permitan realizar las acciones deseadas. Esta información es relativamente fácil de encontrar, incluso dentro de las explicaciones del mismo programa, si se tiene una idea clara de lo que se quiere hacer. El código también se puede encontrar de manera fácil por Internet.

## METEOROLOGÍA

### Objetivo:

Esta función del programa nos proporcionará toda la información, mapas del tiempo y predicciones de los principales servicios meteorológicos de la zona del Mediterráneo catalán y las costas del sur de Francia, además de los principales puertos de España.

### Descripción Funciones:

El primer servicio meteorológico del que sacamos provecho, el Meteocat, nos ofrece las predicciones del estado del mar y el viento en la Costa Brava y en mar abierto. Esta página no presenta mapas, solo unas tablas que dividen las predicciones de mar y viento en intervalos de seis horas durante tres días.

Aparecen diversas tablas, que nos indican el tiempo de las distintas comarcas de la Costa Brava y de las playas situadas entre los cabos más importantes, desde el norte del 'Cap de Creus' 'hasta la 'Punta de Tordera'.

**Predicción**

- Prediccions
- Predicció a curt termini
- Predicció municipal a 8 dies
- Predicció marítima**
- Predicció al Pirineu
- Predicció de calamarsa
- Índex ultraviolat (UVI)

**Models numèrics**

**Avissos de Situació Meteorològica de Perill**

**Dades meteorològiques**

**Climatologia**

**Serveis**

**Divulgació meteorològica**

**El Servei Meteorològic**

**Reiojes de Puñera 1&D En OIU X**

**Comprar Ahora**

**Meteorología**

Inici > Prediccions > Prediccions > Predicció marítima

**Predicció marítima**

Costa Brava    Costa Central    Costa Daurada

Del Cap de Creus a la Punta de la Tordera

- A la costa
- A mar obert

Predicció de la sortida automàtica del model oceanogràfic WAM

Costa Brava: predicció a la costa

del Cap de Creus a la Punta de la Tordera

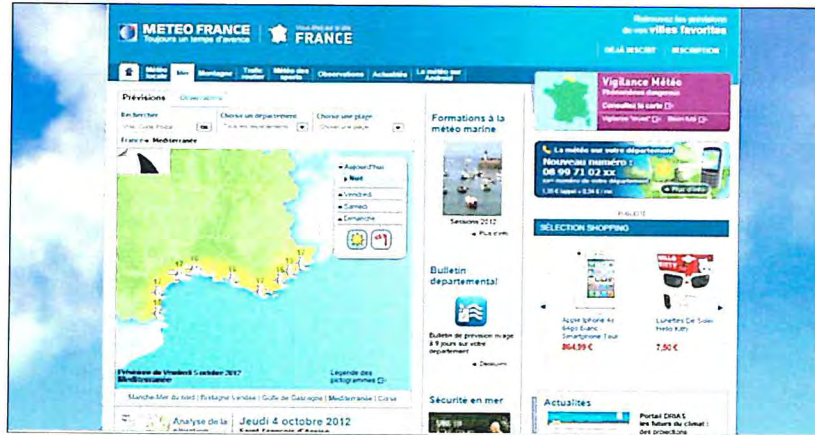
	Dl 04/10		Dy 05/10			Ds 06/10				
	12-18 TU	18-24 TU	00-06 TU	06-12 TU	12-18 TU	18-24 TU	00-06 TU	06-12 TU	12-18 TU	18-24 TU
Mar	Mar arrossada.	Mar arrossada.	Mar entre plana i arrossada.	Mar arrossada.	Mar arrossada.	Mar arrossada.	Mar entre plana i arrossada.	Mar entre plana i arrossada.	Mar arrossada.	Mar arrossada.
Vent	Entre E i SE 2/3.	De SE 2/3 a SW 1/3.	Entre W i SW 1/2.	De SW 1/2 a S i SE 2/3.	Entre S i SE 2/3.	Entre S i SE 1/3.	Entre S i SE 1/2 a var. 1/2.	Var. 1/2 a E 1/3.	Entre E i SE 1/3.	De SE a S 1/3.

Última actualització: 04-10-2012 - 13:00 (hora oficial)  
Propera actualització: 05-10-2012 - 13:00 (hora oficial)

Formato de la página Meteocat

El segundo servicio que encontramos, Météo France, nos ofrece la posibilidad de escoger los núcleos urbanos, playas o zonas de Francia (en nuestro caso es la costa mediterránea) que nos interese conocer las predicciones en tiempo real y de los

próximos tres días (divididos en mañana, mediodía y noche). Esta información aparece en un mapa con los iconos respectivos al tiempo que esté haciendo en cada zona, también se puede escoger el tipo de mapa: de viento, de frentes y de la situación meteorológica.



Formato de la página web de Météo France

El tercer servicio que el programa utiliza es Portus, Puertos del Estado. Este canal presenta un mapamundi interactivo y animado (que puede mostrar las imágenes en formato satélite o con relieve), que ofrece información en tiempo real, predicciones a varios días vista, información histórica y caracterización climática de diferentes parámetros físicos (oleaje, corrientes, viento, etc).



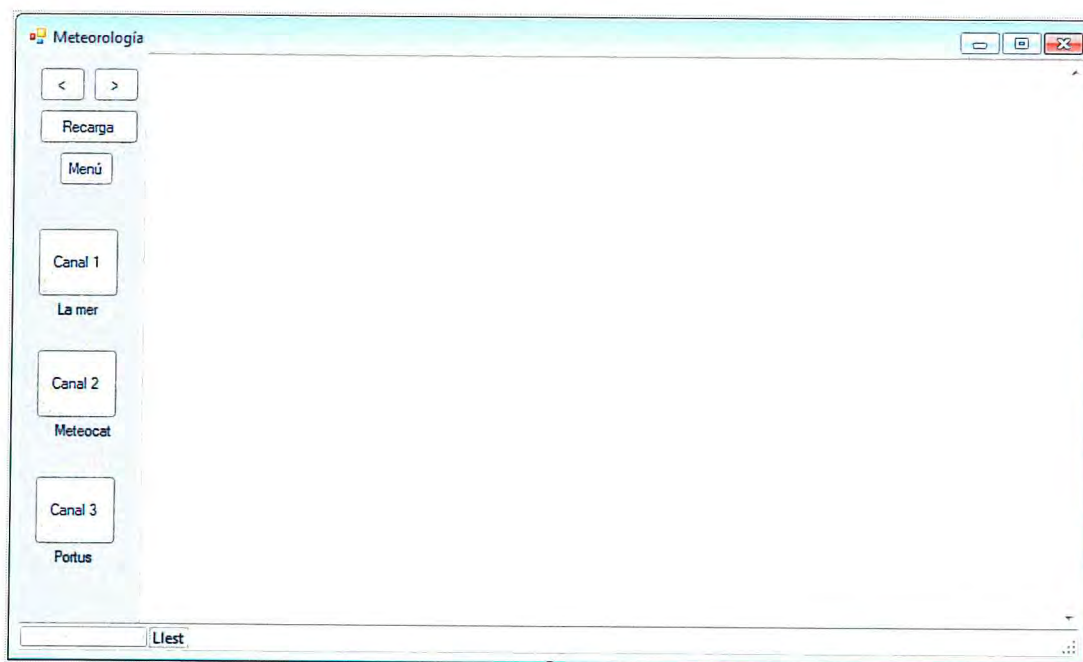
Formato de la página Portus

---

### *Explicación del código:*

Lo que pretendemos con este apartado del programa es hacer uso de las páginas web de servicios meteorológicos detalladas anteriormente para poder obtener la información pertinente sobre las condiciones climáticas de sus zonas geográficas correspondientes.

Este apartado del programa se presenta estéticamente de la siguiente forma:



Los objetos que nos ofrece el programa con los que vamos a trabajar y a construir dicha parte del programa son estos:

- Web browser (navegador)
- Button (botón) x 7
- Status Strip (marcador de estado)
- Progress Bar (barra de progreso)
- Status Label (texto de estado)
- Labels (Etiquetas) x 3

---

La intención es poder construir un navegador adaptado para poder acceder a dichas direcciones URL. Por esta razón vamos a usar el Web Browser porque queremos usar estos servicios que están en la red, pero lo que nos interesa no es navegar por internet, sino visitar los canales informativos de los que sacamos provecho, que son el Servicio catalán Meteocat, el apartado marítimo de la Météo France y el canal Portus. Cada uno de estos servicios tendrá su botón correspondiente que al clicarlo nos dirigirá a la dirección URL de cada servicio. Esto se consigue con la función *CType* y la orden *navigate*:

```
Private Sub Canal1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Canal1.Click
    CType(WebBrowser1, WebBrowser).Navigate("http://france.meteofrance.com/france/mer?MER_PORTLET.path=merprevisionrivage%252FRIV_MEDITERRA")
End Sub
```

Como en cualquier otro navegador, este presenta los clásicos botones de 'volver atrás y ir adelante' para cuando quieras buscar el sitio donde has estado anteriormente y 'recargar la página' para cuando se cuelga el navegador. Pues con la misma función que antes, *CType* y una serie de órdenes respectivas a cada acción deseada lo conseguiremos con solo clicar el botón:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    CType(WebBrowser1, WebBrowser).GoBack()
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    CType(WebBrowser1, WebBrowser).GoForward()
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
    CType(WebBrowser1, WebBrowser).Refresh()
End Sub
```

El navegador es muy plano, y para hacerlo un poco más atractivo a la vista, hemos añadido la barra de progreso y el texto que nos indicará el estado de la web. Al

---

maximizar la ventana de meteorología de nuestro programa, nos quedaban los elementos de estado por en medio de la pantalla, así que para que permanecieran en la parte inferior de la ventana los hemos incluido a través del *Status Strip*. El problema lo hemos encontrado a la hora de hacer funcionar la barra de progreso. No podía cargar los datos si no se le definían unos parámetros mínimos y máximos, por esto le hemos fijado dentro de unos valores con un bucle (i), es decir, que le hemos definido un máximo y un mínimo para que pueda llenarse gradualmente mientras se carguen los datos. La acción del *Status Strip* se iniciará al cambiar de web:

```
Private Sub WebBrowser1_ProgressChanged(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.WebBrowserProgressChangedEventArgs)
Handles WebBrowser1.ProgressChanged

    ToolStripProgressBar1.Maximum = 1
    For i = 0 To ToolStripProgressBar1.Maximum
        ToolStripProgressBar1.Value = i
    Next
    StatusText.Text = WebBrowser1.StatusText

End Sub
```

El texto del objeto *StatusText* está programado para que nos indique las modificaciones del navegador.

Finalmente, para terminar la parte de meteorología del programa, añadiremos un botón cuya función consistirá en volver a la pantalla del menú:

```
Private Sub Button4_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    INA.Show()
    Me.Hide()

End Sub
```

### *Conclusiones:*

El programar un navegador web y sus respectivos componentes no es una tarea que traiga muchas complicaciones. Para un principiante en la programación le será muy fácil encontrar todo tipo de instrucciones y tutoriales por Internet para realizar proyectos semejantes con cualquier clase de lenguaje. El potencial problema de este



---

apartado ha sido programar la barra de progreso. Información tan específica como la que suponía poner en funcionamiento este elemento no se encuentra o no es fácil de encontrar, por lo que se requiere un poco más de experiencia que para programar el resto del navegador.

## **LOCALIZACIÓN**

### *Objetivo:*

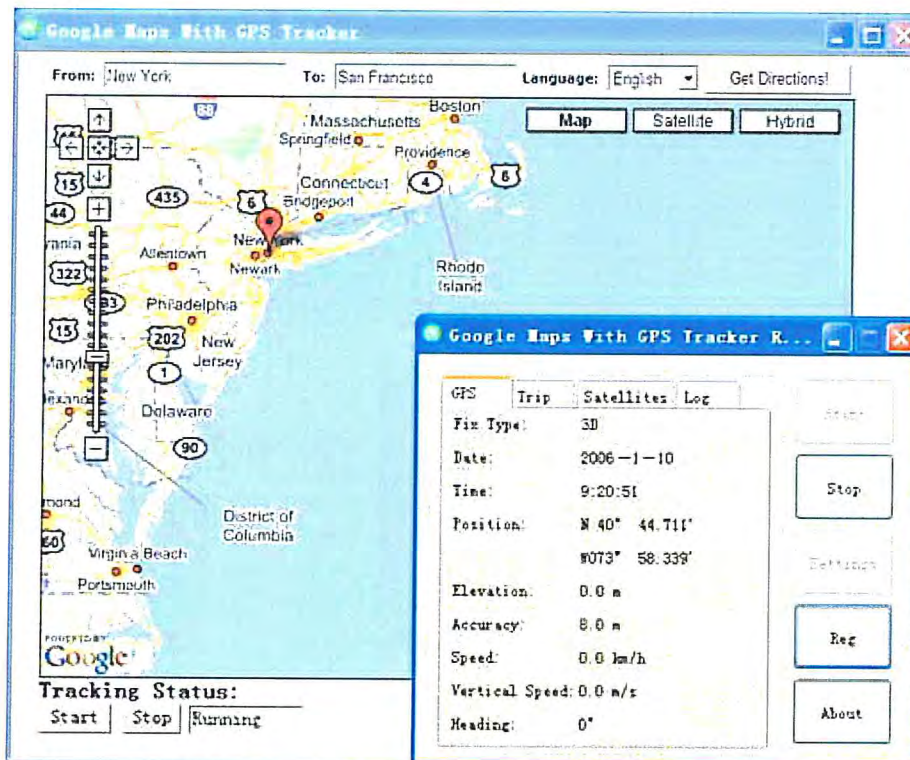
Este apartado del proyecto nos va a servir a modo de herramienta para comunicar con otro programa, el *Google Maps GPS Tracker* (que viene incluido en el disco del programa), que nos mostrará la posición física en tiempo real sobre Google Maps. Esta aplicación dará nuestra posición a través de GPRS o 3G, actualizándola periódicamente.

Para el uso del programa es necesario un dispositivo GPS. En nuestro caso será el *GARMIN GPS 18PC*. Pero puede servir cualquier otro modelo compatible con el ordenador que se use.

### *Descripción del programa GPS:*

Al abrir el programa, aparece un pequeño cuadrado de interconexión con distintas pestañas y botones, está abierto en la pestaña GPS, donde aparecen etiquetas con las magnitudes físicas que nos va a mostrar, como por ejemplo: velocidad, posición , elevación...

La segunda pestaña nos muestra la velocidad instantánea, la distancia recorrida hasta al momento y los grados de tu dirección. Hay la opción de reiniciar los valores que se nos muestran. En cuanto a la tercera y la cuarta, sirven para mostrarnos los satélites a los que estemos conectados o para guardar anotaciones, respectivamente.



*Ventana inicial del GPS y Google Maps mostrando el posicionamiento*

Como habíamos mencionado antes, a la ventana que aparece al iniciar el programa aparecen cinco botones. El primero sirve para iniciar el programa, al clicarlo se empiezan a contar las magnitudes y aparece el *Google Maps* marcando nuestra posición; el segundo para el programa, se detienen las cuentas de las magnitudes y la posición del mapa se queda fija en la última actualización de posición; el tercero abre otra ventana dónde aparecen tres pestañas más: en la primera se nos permite configurar GPS, seleccionar el puerto COM con el que conectaremos con el aparato de GPS y la velocidad de transmisión. La segunda, Garmin Protocolo, permite a los usuarios de GPS Garmin introducir su ID de producto, versión de software, y otros datos, así como establecer el protocolo. La pestaña Compartir contiene un único botón que nos permite compartir la ubicación instantánea a través de un sitio Web. El cuarto botón sirve para registrar la versión ampliada del producto en el caso de compra; y el quinto nos da información más detallada sobre el programa.

---

### *Explicación del código:*

Esta parte del programa solo nos sirve para conectar con el programa de GPS, con lo cual, el único elemento del que hacemos uso es un botón que nos hará regresar a la pantalla menú.



Al pulsar el botón 'Menú', se cierra la ventana actual y se abre INA, la inicial.

```
Private Sub Button4_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    INA.Show()
    Me.Hide()
End Sub
```

### *Conclusiones:*

El resultado final de este apartado no se parece a la idea inicial, en la que se había planeado manejar el programa GPS desde dentro de nuestro proyecto para poder interactuar de manera más amplia con el resto del programa. El mayor problema es que la programación con *Visual Basic* es bastante limitada al tratar con otros programas, no existen bibliotecas ni lenguaje específico que permita incluir un programa independiente dentro de nuestros proyectos. Hemos tenido que recurrir a una solución improvisada para poder usar las funciones ofrecidas por *Google Maps GPS Tracker* y poder, a la vez, volver a la pantalla de menú de nuestro programa.

## **COMUNICACIÓN**

### *Objetivos:*

El apartado de comunicación está dedicado a facilitar el contacto a distancia entre barcos, a la vez que dar a conocer y recuperar el uso de un lenguaje como es el código morse, aumentando la velocidad de emisión y traducción.

---

### *Introducción al Código Morse:*

El Código Morse es un medio de comunicación basado en la transmisión y recepción de mensajes empleando sonidos o rayos de luz y un alfabeto alfanumérico compuesto por puntos y rayas.

Dicho lenguaje muestra una clara desventaja respecto a la radiodifusión mediante voz, ya que la velocidad de escritura es muy lenta en comparación, cosa que dificulta enviar y traducir mensajes extensos.

Pero su empleo es perfectamente utilizable hoy en día cuando la existencia de condiciones atmosféricas adversas no permitan el empleo de otros medios más desarrollados como, por ejemplo, la transmisión de voz.

Además este código es internacional, eso elimina los problemas entre idiomas, los mensajes claves como el "SOS" ("... --- ...") son mundialmente conocidos. Como lenguaje reconocido internacionalmente, tiene ciertos patrones que hace falta respetar para poder enviar un mensaje de forma apropiada, así como ciertos códigos paraverbales para facilitar las transmisiones.

### *Descripción de funciones:*

En nuestro programa lo que hacemos es reducir la dificultad y duración de la ardua tarea de escribir símbolo a símbolo toda una frase para luego enviarla como morse para que sea otra vez traducida por el receptor. Hemos confeccionado un traductor instantáneo que con solo un clic nos convertirá cualquier letra, palabra o frase escrita en mayúscula al abecedario en morse y viceversa. Además, incluye la opción de escuchar el conjunto de signos del abecedario morse.

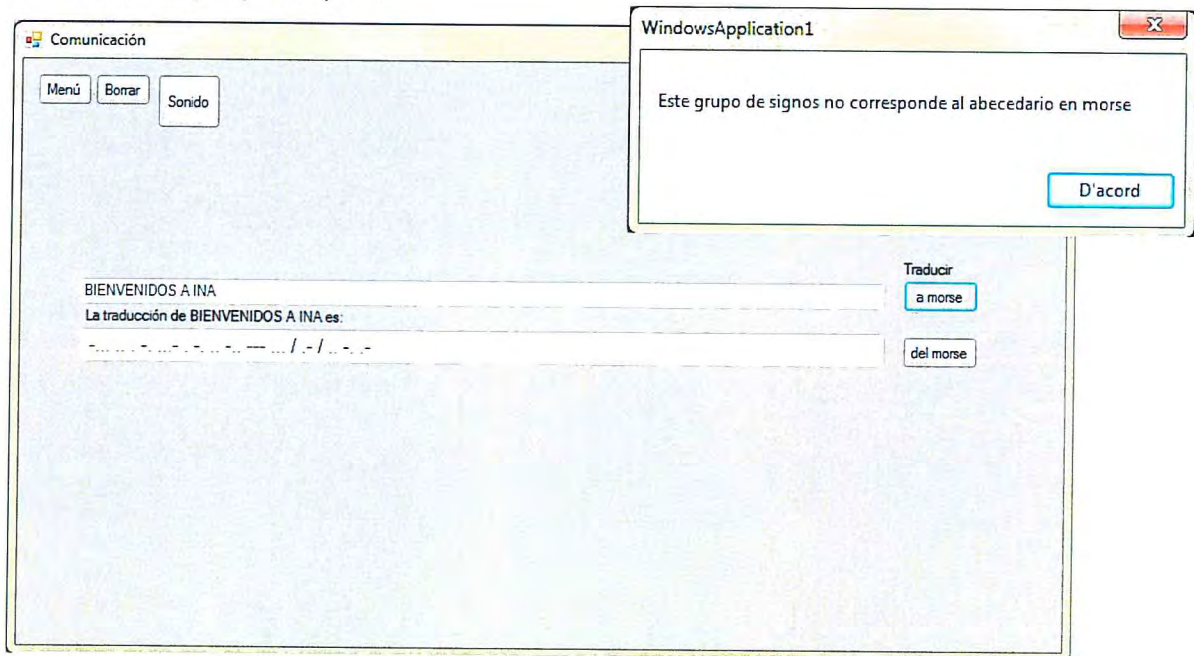
En este apartado incluiremos, también, un soporte hardware para exteriorizar el sonido producido por el programa.

### *Explicación del código:*

Para realizar el traductor de morse, vamos a necesitar los siguientes elementos y los vamos a distribuir de la siguiente forma para mejor estética:

- Botón x 5

- Text Box (cuadro de texto) x 2
- Message Box (ventana emergente)
- Labels (etiquetas) x 3



Empezaremos hablando de los dos cuadros de texto que se sitúan en la parte central de la ventana – en el superior se escriben letras en mayúsculas y espacios, en el inferior se escriben los puntos, rayas y espacios correspondientes al abecedario en morse –, cada uno con un botón a su lado.

Como indica la etiqueta situada encima de los botones, estos sirven para traducir a morse y del morse respectivamente. Primero explicaremos como realizar la traducción a morse. Para esto tendremos que definir dos cadenas: la primera que contenga las letras del abecedario y el espacio, la segunda que contenga los grupos de puntos y rayas correspondientes a cada letra ordenados alfabéticamente. El espacio entre palabras lo hemos definido como "/" para facilitar la lectura de los textos.

Una vez definidas las variables, lo que le estamos diciendo al programa, es que lea lo que está escrito en el primer cuadro de texto y que para cada carácter que encuentre

vaya a buscar sus puntos y rayas correspondientes a la segunda cadena, dónde estos están definidos. Luego, que en el segundo cuadro de texto haga aparecer toda la cadena de letras del primer cuadro, pero sustituidas por los símbolos que les corresponden.

```

Public Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    Advertencia.Text = ""
    Dim noms0 As String() = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
"K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "
"}
    Dim noms1 As String() = {".-", "-...", "-.-.", "-..", ".", ".-.-", "-.-.",
"....", ".-", ".-.-", "-.-.", ".-.-", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "/"}
    Dim sortida As String
    Dim k As Integer
    Try
        For k = 0 To (Len(TextBox1.Text) - 1)
            Dim ret As String = TextBox1.Text(k)
            Dim ret2 As Integer = Array.IndexOf(noms0, ret)
            sortida = sortida & noms1(ret2)

        Next
        Traducció.Text = "La traducción de " & TextBox1.Text & " es: "
        TextBox2.Text = sortida
    Catch ex As Exception
        MsgBox("Se tienen que escribir letras del abecedario en mayúsculas")
    End Try
End Sub

```

La traducción de morse a letras es un tanto más complicada que la anterior. Antes hemos definido unas cadenas, una con las letras del abecedario y otra con los puntos y rayas correspondientes a cada letra. Lo que hacíamos era coger un símbolo de la primera cadena y substituirlo por sus símbolos correspondientes de la segunda cadena. Pero el problema ahora es que los caracteres de la segunda cadena están compuestas por más de un símbolo, y nosotros solo podemos utilizar uno. Pues lo que vamos a hacer con las siguientes líneas de código es decirle al programa que cuando lea un espacio unifique todos los símbolos colocados detrás de este y que busque en la segunda cadena un grupo de puntos y rayas que sea igual, en el caso de que lo encuentre, que lo sustituya por la letra correspondiente de la primera cadena y lo escriba en el cuadro de texto indicado.

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click

    Advertencia.Text = ""
    Dim noms3 As String() = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K",
"L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", " "}
    Dim noms2 As String() = {".-", "-...", "-.-.", "-..", ".-", ".-.-.", "-.-.",
"....", ".-", ".-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.",
"-", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "/"}
    Dim sortida2 As String = ""
    Dim ret2 As String = ""
    Dim k As Integer = 0
    Dim j As Integer = 0
    Dim w As Integer = 0
    Dim ret5 As String = " "
    Try
        While k < (Len(TextBox2.Text))

            While TextBox2.Text(j) <> " "
                ret2 = ret2 & TextBox2.Text(j)
                j = j + 1
                w = w + 1
            End While

            k = k + w

            Dim ret3 As Integer = Array.IndexOf(noms2, ret2)
            sortida2 = sortida2 & noms3(ret3)
            j = j + 1
            ret2 = ""
            w = 0
            k = k + 1

        End While

        Traducció.Text = "La traducción de " & TextBox2.Text & " es: "
        TextBox1.Text = sortida2

    Catch ex As Exception
        MsgBox("Este grupo de signos no corresponde al abecedario en morse")
        Advertencia.Text = "Atención! Es necesario escribir un espacio al final
de cada grupo de puntos y rayas correspondientes a una letra"
    End Try

End Sub

```

La función *Try/Catch* nos detecta los errores, en este caso a la hora de traducir, y nos hace aparecer una ventana que nos avisa de que algo no va bien. Esto evita que el programa se bloquee o que la depuración se ature para avisar-nos de que algo no va bien. En nuestra ventana emergente nos aparece una advertencia que nos prevé de

---

los errores más comunes en cada caso: en el primero nos dice que el contenido del cuadro de texto tiene que estar formado solo por, letras escritas en mayúsculas. En el segundo caso nos informa que lo escrito en el segundo cuadro de texto no corresponde con el abecedario morse, además nos hace aparecer una etiqueta de advertencia debajo de este cuadro.

En este apartado aparecen dos etiquetas programadas, la primera se llama *Traducció*, y lo que hace es indicarnos el texto que acaba de ser traducido. La segunda se llama *Advertencia* y su función es advertirnos, cada vez que no se pueda traducir del morse, que se tiene que escribir un espacio al final de cada grupo de puntos y rayas correspondientes a una letra, ya que este es necesario y puede ser un error frecuente a la hora de traducir.

Los botones de tamaño más reducido que aparecen en la parte superior izquierda de la ventana tienen las siguientes funciones, por orden de izquierda a derecha: el primero es el mismo que se encuentra en las otras partes del programa, y su función es devolvernos a la pantalla inicial. El segundo es propio de este apartado, su función es borrar todo lo escrito en la ventana, tanto el contenido de los cuadros de texto como las etiquetas de traducción y de advertencia:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Borrar.Click
    TextBox1.Text = ""
    TextBox2.Text = ""
    Traducció.Text = ""
    Advertencia.Text = ""
End Sub
```

El tercero es el encargado de hacer que el conjunto de puntos y rayas se puedan escuchar. Para ello habrá que llamar a la tarjeta de sonido (*kernel32*) y declarar la función *beep*, a la que se le va a definir frecuencia y duración entre paréntesis.

```
Public Declare Function Beep Lib "kernel32" (ByVal soundFrequency As Int32, ByVal soundDuration As Int32) As Int32
```



---

La frecuencia que hemos escogido para nuestro programa es la de 1000 Hz y la durada variará, por el punto será de 150 ms y por la raya 450 ms (correspondiendo con las duradas establecidas por ley).

Al hacer clic en el botón de sonido se iniciará la acción, en la que el programa leerá lo del primer cuadro y lo irá sustituyendo por sus símbolos correspondientes. Pero esta vez, no los escribirá en el segundo cuadro sino que los reproducirá. Para hacerlo tendremos que definir el sonido de cada letra, por ejemplo el de la letra A:

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Altaveu.Click

    Advertencia.Text = ""
    Dim noms0 As String() = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
"K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "
"}
    Dim noms1 As String() = {".- ", "-... ", "-.-.", "-.. ", ". .", ".-.- ",
"-.-.", "-... ", ". .", ".-.-.", "-.-.", "-... ", "-.-.", "-.-.", "-.-.",
"-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.", "-.-.",
"/ "}
    Dim k As Integer

    For k = 0 To (Len(TextBox1.Text) - 1)
        Dim ret As String = TextBox1.Text(k)
        Dim ret2 As Integer = Array.IndexOf(noms0, ret)

        If ret = "A" Then
            Beep(1000, 150)
            Beep(1000, 450)
            Beep(0, 150)
            .
            .
            .
        ElseIf ret = "/" Then
            Beep(0, 150)

        End If
        Beep(0, 1200)
    Next
```

Como podemos ver, después de terminar de leer las letras hay un beep de frecuencia 0 y 1200 de duración. Esto actúa como freno del procesador del ordenador, ya que al leer los distintos símbolos que pudieran haber escritos en el cuadro iba demasiado rápido para que la tarjeta de sonido pudiera reproducirlos a tiempo. El resultado de

---

este descompás era un conjunto de pitidos al azar producidos por una tarjeta saturada de información. Al indicar la función beep final, el procesador ha quedado ralentizado, cosa que permite escuchar los pitidos que realmente deberían sonar.

### *Construcción del altavoz:*

Durante las dos últimas semanas del proyecto, nos hemos dedicado a la construcción de un soporte hardware para el programa. La idea de este es cumplir con la misión de exteriorizar el sonido producido en el traductor de morse, ya que el ordenador suele estar en los interiores del barco y sus altavoces no son lo bastante potentes. Este altavoz es solo una muestra a pequeña escala de lo que sería un repetidor de cubierta, que enviaría los señales producidos por el programa a un ratio bastante extenso.

Consultar los anexos para ver los detalles de funcionamiento y construcción.

### *Conclusiones:*

Esta ha sido sin duda la parte más complicada y trabajada del proyecto, tanto por el apartado de software como por el altavoz construido a modo de soporte de salida del programa. Para un principiante en la programación como yo, realizar esta parte del proyecto ha traído más trabajo del aparente. El traductor de morse es un nivel, a mi parecer, alto en este tipo de lenguaje y, es obvio, que no hay ninguna función específica del programa para realizar la acción de traducir y, mucho menos, información en Internet. La clave para llevarlo a cabo ha sido coger experiencia y familiarizarme con distintas funciones del lenguaje mediante la creación de pequeños proyectos específicos para cada función, que no han sido incluidos en el proyecto. La autoformación realizada a partir de las exigencias que requería el llevar a cabo un proyecto de esta envergadura cumple con creces uno de los objetivos previos al empezar el programa, el iniciarme en el mundo de la programación informática.

En cuanto a los problemas del hardware, no fue fácil encontrar la forma de hacer interactuar un elemento externo al ordenador con el programa de Visual Basic. La información encontrada en tutoriales y proyectos ajenos por Internet me hizo ver que

---

sin librerías el VB no puede contactar con los puertos COM, y extraer la información por el usb requiere el uso de circuitos integrados PIC previamente programados con otro tipo de lenguaje.

## 5. Conclusiones generales

Una vez terminado el programa, podemos asegurar que la idea previa de este era bastante distinta al trabajo conseguido. Por ejemplo, lo que pretendíamos hacer en el apartado de comunicación era una herramienta que pudiera captar el sonido exterior, pasar el sonido entrante a puntos y rayas y estos a letras. Para realizar un proyecto de este calibre se necesitan librerías como BASS.NET, que requieren un nivel de conocimientos en programación superior al de un principiante.

Respecto a INA, podemos decir que hemos conseguido hacer una herramienta de bastante utilidad y que hemos realizado un programa relativamente completo, pero con algunas limitaciones para utilizarlo en la vida real en alta mar. Por ejemplo, al utilizar un lenguaje que no es compatible con el manejo de puertos USB, los altavoces necesarios para una embarcación que utilizara el programa no podrían conectarse inalámbricamente con el ordenador; el cableado empleado podría dificultar la logística del barco, y en el caso que se interiorizasen dichos cables, se requerirían unas reformas de costos muy elevados. Sin embargo los objetivos planteados respecto a este Programa de Navegación han sido cumplidos de forma satisfactoria.

Por lo general podemos afirmar que el Visual Basic es un lenguaje relativamente fácil de usar, al ser muy intuitivo permite aprender rápido a manejar el programa e iniciarse en la programación informática, por lo tanto lo podemos definir como el programa de aprendizaje definitivo. Pero también hace falta hacer un inciso en el hecho que el VB es un programa bastante limitado, depende de complementos y librerías cuando intentas hacer proyectos de cierto nivel.

Este ha sido, por encima de todo, un trabajo de autoformación. Por esta razón, aunque el esfuerzo realizado para llevar a cabo el proyecto y el número de horas dedicadas al

---

aprendizaje de códigos, búsqueda de informaciones y de familiarización con la programación han sido considerables, el resultado no las refleja en su totalidad.

La formación conseguida a partir de esta dedicación ha significado el cumplimiento del mayor de los objetivos, ahora que conozco las bases de la programación informática se me abren las puertas para realizar otros proyectos más ambiciosos e, incluso, con un lenguaje distinto al empleado.

---

## 6. Bibliografía

Las siguientes páginas web son los tutoriales y guías de aprendizaje que he seguido, en gran parte del trabajo, para realizar el proyecto. Las fechas no se especificarán porque han sido utilizadas en repetidas ocasiones a lo largo del trabajo:

[http://msdn.microsoft.com/es-es/library/w032hx3c\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/w032hx3c(v=vs.80).aspx)

<http://www.recursovisualbasic.com.ar/htm/tutoriales.htm>

<http://www.solovb.net>

<http://personales.upv.es/jpgarcia/LinkedDocuments/macrosVisualBasicParaExcel.pdf>

A estas webs se suman tantas otras que me han servido para realizar pequeños proyectos para familiarizarme con el lenguaje, pero no han sido incluidas porque no guardan más relación que la esmentada con nuestro programa.

También se utilizó el libro *Manual imprescindible de Microsoft Visual Basic 2008* de Juan Diego Gutiérrez Gallardo

Para conseguir la información del código Morse han sido utilizadas las siguientes páginas:

[http://ca.wikipedia.org/wiki/Codi\\_Morse](http://ca.wikipedia.org/wiki/Codi_Morse)

[http://www.asifunciona.com/tablas/codigo\\_morse/codigo\\_morse\\_1.htm](http://www.asifunciona.com/tablas/codigo_morse/codigo_morse_1.htm)

[http://es.wikipedia.org/wiki/C%C3%B3digo\\_Q\\_de\\_se%C3%B1ales](http://es.wikipedia.org/wiki/C%C3%B3digo_Q_de_se%C3%B1ales)

[consultadas el 6 de octubre del 2012]