

Módulo 1

Valores de un buen programador

Tecnara – Cursos de formación



Índice de contenidos

1. **INGLÉS, INGLÉS y más INGLÉS**
2. Espíritu crítico y explorador: Practicar
3. Dudas e información, primero en la Web
4. Paciencia y ánimo al encontrar errores desconocidos
5. Documentar el código
6. Organización/modularización del código
7. Creer en la magia

01

INGLÉS, INGLÉS y más INGLÉS

INGLÉS, INGLÉS y más INGLÉS

- El mundo de la programación hace uso predominantemente del inglés, y C# no es la excepción:
 - Los nombres dados a las cosas (ya iremos viendo qué 'cosas' con nombre existen) que vienen instaladas por defecto para C# están todos en inglés
 - Los nombres de las cosas ya hechas que no vienen por defecto en C# pero que se pueden descargar de Internet, también están en inglés

INGLÉS, INGLÉS y más INGLÉS

- Se recomienda escribir el código usando nombres en inglés siempre que resulte cómodo, para coger práctica
 - Sin embargo, no se obliga: si alguien entiende mejor su propio código escribiendo en su lengua nativa, puede hacerlo
 - Quien decida escribir programas usando nombres en un idioma distinto al inglés, deberá tener en cuenta que no podrá utilizar para los nombres caracteres idiomáticos que no se usen en el alfabeto anglosajón
- Ejemplo: en español, no podrá programar escribiendo nombres con tildes o eñes...

02

Espíritu crítico y explorador: Practicar

Espíritu crítico y explorador: Practicar

- No quiero que NADIE se crea NADA solo porque yo, como profesor, lo diga
- Quiero que lo COMPROBÉIS todo por vosotr@s mism@s
- Creédme, es la mejor forma de aprender programación
- Se recomienda jugar con los programas, como se propuso con el Hello World!

Espíritu crítico y explorador: Practicar

- Incluso aunque ya estén hechos y funcionen bien, tratad de:
 - Pensar en formas de mejorarlos, o de adaptarlos a vuestro gusto
 - Cambiarlos para que hagan lo mismo con instrucciones diferentes, si las conocéis
 - Y si no las conocéis, pero creéis que pueden existir, tratar de buscarlas y aprenderlas
 - Añadirles o quitarles cosas para ver qué pasa
 - Descubrir qué tipos de errores pueden producirse en vuestros programas
 - Forzar el código para que tenga errores de compilación o de ejecución
 - Entender cómo avisa Visual Studio de esos errores
 - Tratar de entender los mensajes que muestra VS indicando esos errores, para poder identificarlos cuando os salgan con errores de verdad (no esperados)

03

Dudas e información, primero en la Web

Dudas e información, primero en la Web

- Han existido muchos programadores de .NET Framework antes que vosotros
- Esos programadores han tenido que escribir muchos programas iguales o muy parecidos a los que escribiréis vosotros
- Mientras los escribían, se han encontrado con el 99,9999999...% de los errores que os vais a encontrar vosotros

Dudas e información, primero en la Web

- Una gran cantidad de almas caritativas entre esos programadores, han decidido publicar en páginas de Internet los problemas que se han encontrado, junto con las soluciones que han adoptado.
- Además, existen muchos foros virtuales de acceso público especializados en programación, donde:
 - un usuario registrado puede solicitar ayuda a la comunidad de programadores que participen en ese foro, preguntando cómo solucionar un error o cómo llegar a un resultado determinado y obteniendo respuestas a sus problemas
 - Cualquier internauta random puede consultar esos foros y enterarse del problema planteado y las soluciones ofrecidas, gratuita y rápidamente

Dudas e información, primero en la Web

- Conclusión: TODAS las respuestas que necesitéis ya están en Internet
- Pero para acceder a ellas, hay que saber qué escribir en el buscador
 - Esto también se aprende equivocándose mucho al principio
- Se encuentra mucha más información sobre cualquier tema relacionado con la programación si se busca en inglés que si se busca en otro idioma

Dudas e información, primero en la Web

- Si tras un tiempo buscando en la web, no se encuentra la respuesta, o si directamente no se ha entendido algo y no se sabe ni siquiera qué escribir en el buscador para llegar a la respuesta, ENTONCES SÍ, pedidme ayuda a mí, al profesor
- ‘Se supone’ que yo estoy como último recurso omnisciente (que lo sabe todo, de ahí el ‘se supone’ 😊) al que debéis recurrir si:
 - Estáis bloqueados y no sabéis por dónde tirar (muy común al principio)
 - Creéis saber cómo se solucionaría un problema, pero no sabéis qué escribir en la web para buscar eso que pensáis (común más adelante)

Dudas e información, primero en la Web

- Por mucho que uno tenga un carácter resolutivo y un orgullo que le haga sentirse obligado a resolver sus dudas por sí mismo, pensad que estamos en un curso, es decir, en un entorno adaptado para el aprendizaje dirigido
- Si pasa una notable cantidad de tiempo y la duda sigue sin resolverse, es conveniente tragarse ese orgullo y pedir ayuda a otro compañero (lo cual puede ser beneficioso para los dos), o a mí, al profesor ‘omnisciente’

04

Paciencia y ánimo al encontrar errores desconocidos

Paciencia y ánimo para resolver errores

- Hace falta tener paciencia y no desanimarse cuando aparece un error (sea de compilación o de ejecución) que no se sabe dónde aparece o cómo se soluciona, ni siquiera con la información sobre el error que nos brinda Visual Studio
- A veces, pasamos mucho tiempo revisando un trozo de código porque pensamos que el error está ahí, pero el error está en otro sitio
- A veces, el error está todo el rato DELANTE DE NUESTROS OJOS, pero no nos fijamos en él porque es algo tan básico que damos por hecho que no lo hemos podido escribir mal...

Paciencia y ánimo para resolver errores

- Igual que con las dudas y las búsquedas, con la resolución de errores hay un tiempo para que lo intentéis vosotr@s, pero pasado ese tiempo, si no habéis avanzado nada, preguntad.
- Es curioso cómo a veces la forma más efectiva para encontrar un error que llevas mucho tiempo sin localizar pasa por dejarlo, pasar a hacer otra cosa, despejarte la cabeza, y luego retomarlo con la cabeza despejada.
- Al tener la mente despejada, el cerebro está más abierto a comprobar cosas que antes, cuando estaba saturado, daba por supuestas para no forzarse tanto

05

Documentar el código

Documentar el código

- Además del código que se traduce y luego se ejecuta en el ordenador, todos los lenguajes de programación permiten escribir líneas que no se van a traducir ni ejecutar, donde se puede escribir lo que se desee
- Estas líneas se pueden utilizar para explicar, en el idioma deseado, lo que se supone que hace el código cercano
- Aquí no se recomienda el inglés, sino el idioma que mejor entienda cada uno, normalmente su lengua natal...

Documentar el código

- Hay que pensar que esa documentación la pueden leer otras personas
 - En una organización con un idioma común mayoritario, normalmente se pide a todos los desarrolladores que documenten usando ese idioma mayoritario
 - En otras organizaciones con proyección internacional, se puede aplicar la política de empresa de documentar código siempre en inglés
 - En este curso, yo, como profesor, leo bastante bien en inglés y español, y me semi-defiendo con el francés, pero hasta ahí. Solo admito esos tres idiomas para documentar código

Documentar el código

- Conviene tener en la cabeza estos tres puntos al ir a documentar código:
 - Debe servirnos para poder entenderlo nosotros ahora
 - Debe ser igual de útil para poder entenderlo nosotros si lo leemos dentro de un año, cuando ya no nos acordemos de qué valor contenía cada variable simplemente mirando su nombre
→ Para eso, además de documentar correctamente, es MUY RECOMENDABLE elegir para las variables y funciones nombres AUTO-EXPLICATIVOS de lo que hacen o lo que contienen, aunque eso los convierta en nombres muy largos
 - Debe servir para que lo puedan entender más o menos rápidamente otras personas que no hayan visto ese código antes en toda su vida
→ de nuevo, esto es más fácil si además de la documentación se usan nombres AUTO-EXPLICATIVOS

Documentar el código

- Al principio igual es interesante documentar al detalle, casi línea por línea, para grabar en nuestra cabeza lo que va haciendo el programa (al escribirlas, las ideas se fijan mejor en el cerebro)
- Más adelante, conforme nos acostumbremos al código, podremos documentar con menor detalle, explicando la funcionalidad a nivel general

06

Organización/modularización del código

Organización/modularización del código

- En vez de escribir programas de 500 líneas seguidas de código, es preferible separar el código en distintas funciones que se invoquen unas a otras, y que cada una no tenga más de (en torno a, alrededor de) 20 líneas
- Esto facilita la visualización y comprensión del código, la localización de errores, y permite la REUTILIZACIÓN de funcionalidades (trozos de código que cumplen una función particular)

07

Crear en la magia

Crear en la magia

- A mí no me tenéis que creer nada de lo que diga, pero al funcionamiento del programa que ejecutéis, sí
- La mayor parte del código de un programa no lo hemos escrito nosotros, sino que son llamadas a funciones de biblioteca que ha escrito otro, y no siempre (de hecho lo habitual es que no) es posible tener acceso al código C# de las funciones a las que invocamos desde nuestro programa
 - Las bibliotecas básicas de C#, las bibliotecas adicionales ya instaladas de .NET Framework o las bibliotecas disponibles en Internet, vienen compiladas en archivos .dll, que no tienen ni una sola línea escrita en C# (normalmente están escritas en Lenguaje Intermedio)

Crear en la magia

- Normalmente, para esas funciones de biblioteca que no hemos escrito nosotros, existe una documentación de su funcionamiento que podemos consultar en alguna web
- Esa documentación informa de:
 - Una descripción de lo que hacen si la ejecución es correcta
 - Qué datos de entrada necesitan para ejecutarse sin errores
 - Qué datos de salida nos devuelven
 - Qué excepciones pueden generarse si hay algún error durante su ejecución
- Habitualmente, esa información suele ser suficiente para resolver errores relacionados con una determinada función que no hayamos escrito nosotros

Creer en la magia

- Pero hasta ahí: Si estamos interesados en detalles internos de una determinada función que nos provocan mucha curiosidad, habitualmente nos tendremos que quedar con las ganas de conocerlos
- Como último recurso, si la curiosidad no nos deja dormir por las noches, podemos intentar acceder al código original solicitando trabajo en Microsoft o aplicando técnicas de ingeniería inversa
 - Pero como ambas opciones quedan fuera del alcance de este curso, creed en la magia: Funciona. ¿Cómo? No sé, pero funciona... así que a otra cosa...

MUCHAS GRACIAS



Pasión por la innovación

www.integratecnologia.es