

Como ya os he comentado anteriormente he previsto realizar con vosotros un ejercicio de aprendizaje de vital importancia en estos nuestros primeros pasos.

Este ejercicio es completamente opcional sobre todo en los perfiles más avanzados, si alguien considera que no le va a aportar nada que lo indique. En cualquier caso, os animo a resolverlo pues para ellos su resolución será relativamente sencilla. Los que prefieran no realizar este programa pueden optar por realizar otro tipo de formación complementaria, deberéis indicar esto al inicio para poder asignaros la formación.

He diseñado este ejercicio por varios motivos, el primero es eliminar todo tipo de dependencias para que podáis ser vosotros los que analicéis la solución, la desarrolléis y os deis cuenta de los problemas que origina un software sin calidad, el primer objetivo es entender porque es relativamente fácil que todo se complique sobre todo cuando un sistema escala, es muy habitual que en una aplicación los problemas surjan cuando las aplicaciones crecen. Así pues, mi primer objetivo es haceros ver como de fácil es cometer errores y que algo inicialmente sencillo se torne complicado, trato de enseñaros que los problemas que yo veo en vuestro código están ahí por una razón y este motivo no esta vinculado a decisiones de otros, soluciones impuestas por terceros u otros aspectos. Para mí es importante conocer a cada uno de vosotros, no voy a

valorar vuestra experiencia, el número de años que lleváis aquí o las horas que pasáis en la oficina, mi único objetivo es ver el interés por acometer nuevos retos porque necesito identificar a aquellos perfiles a los que les gusta más investigar, adoptar nuevas tecnologías, etc. A los demás tampoco debe preocuparos, la compañía necesita diversidad y también personas que se sientan cómodas con su trabajo diario por lo que este ejercicio no supondrá ninguna barrera.

Se que estáis cansados y que vuestro día a día es, a veces, muy complicado, así que os pido que me deis la oportunidad de mostraros todo lo que este ejercicio puede aportaros, necesito que logréis tener mi visión y para convenceros es imprescindible que hagáis este esfuerzo.

Necesitareis al menos una hora diaria para su resolución aquí en Semicrol pero también advierto que dependiendo del nivel de cada uno es muy posible que preciséis tiempos externos complementarios para su resolución, porque si no se puede demorar mucho. El tiempo ahora no es relevante, si lo es, que no os rindáis, que lo intentéis una y otra vez hasta conseguir resolverlo, es probable que en este recorrido os canséis y penséis en abandonar, os animo a que no lo hagáis, todo esfuerzo siempre tiene recompensa.

No se trata de completarlo perfectamente, se trata de **aprender**, de **esforzarse**, de **rehacerlo** una y otra vez hasta que lo consigáis para que os deis cuenta de que la solución no es un problema de la compañía o de las decisiones de terceros, **la solución está en cada uno de nosotros.**

Este proyecto requiere de un aprendizaje y un alineamiento coordinado y por ello nos facilitará el proceso de aprendizaje:

## Reglas

- El lenguaje, la plataforma o cualquier otra decisión la tomáis vosotros, me da igual que uséis java, javascript, php, rust, go, ruby, python, c++, lo que queráis, lo que vais a aprender es independiente de la tecnología. Mi consejo es c# porque lo manejaís a diario y tiene dentro del framework de .net muchas librerías y funciones que os pueden ayudar en su resolución.
- Tiene que poder ser probado, a través de un interfaz, de un programa de consola o de cualquier medio que consideréis apropiado, pensarlo bien, lo decidís vosotros.
- Os animo a que si el inglés no supone un gran problema intentar resolverlo en este idioma.
- Se puede abandonar cuando se quiera, si no os gusta ni os motiva, si creéis que nos os aporta ningún valor, dejarlo. Esto es por y para vosotros, por lo tanto, sois vosotros los que tomáis la decisión.
- Podéis buscar información para resolverlo, copiar el código de alguna aplicación similar y adaptarlo, no me importa, pero tener en cuenta que esta es la primera parte, el programa va a escalar y quizás os encontréis con algún problema que otro software no es capaz de resolver.
- Pretendo que el ejercicio sea individual, pero si estimáis hacerlo en un equipo de dos personas adelante. De hecho, vamos a implementar una nueva herramienta nativa de Visual Studio llamada Microsoft Live Sharing para que podáis interactuar los unos con los otros. Es interesante pues la adoptaremos en un futuro para la programación por parejas. También podéis usar Teams para esto.
- Yo estaré a vuestra disposición, podréis llamarme para aclarar cualquier duda y ayudaros a resolverlo, comentarlo en la oficina o pedir ayuda a otro compañero. No voy a deciros como resolverlo, tan solo os diré como podéis mejorar vuestro diseño haciendo de guía, os diré que debéis aprender o estudiar para poder avanzar, así que todo el esfuerzo le haréis vosotros.

Antes de abandonar, es **mejor pedir ayuda**, no debe preocuparos, ni quiero que os desmotivéis, en mi experiencia todo el mundo que se esfuerza, puede realizarlo, incluso personas sin un alto conocimiento del Framework y del lenguaje.

# Un reto por aprender CALIDAD

- Si queréis realizarlo o no, es asunto solo vuestro, pero si os decidís por realizar este ejercicio no hagáis trampas, es un ejercicio de aprendizaje, es fácil copiar a un compañero, pero esto no os aportara nada.

## Objetivos del ejercicio.

El objetivo principal es aprender, no hacer un programa para ver quien es el mejor, el comienzo desde luego es relativamente sencillo y en las fases posteriores se irá complicando, os aconsejo pensarlo bien, analizarlo, no os voy a decir como tiene que funcionar, tendréis que descubrirlo vosotros, descomponer los distintos problemas antes de escribir ni si quiera la primera línea de código.

No será fácil, lo importante no es si llegáis a resolverlo o no, es el camino que vais a recorrer y lo que vais a aprender para hacerlo.

- El ejercicio; en su planteamiento inicial, desvela parte de su alcance, por lo que espero de vosotros que **planifiquéis su posible escalabilidad en todos aquellos aspectos que un buen programador ha de estimar y prever.** De lo contrario, puede complicarse en gran medida, deberéis plantearos este ejercicio como un reto.

- El propósito es que lo resolváis de **forma autónoma**, pero habrá necesidades de interacción conmigo. Por supuesto estaré siempre a vuestro lado para ayudaros y conocer vuestros avances y dificultades.
- Deseo conoceros en el ambiente de programación para **aportaros principios y buenas prácticas** tan necesarias que algunos ya utilizan.
- Algunos perfiles más avanzados tienen ventaja, pues conocen algunas técnicas y formas de resolverlo, en cualquier caso, me parece interesante para ellos abordarlo, por el camino quizás aprendan como mejorar en algún aspecto, los primeros que logren finalizarlo podrán **ayudar a los demás si lo necesitan.**
- Aprender como la **complejidad** de un programa surge normalmente **cuando escala.**
- Aprender como **si no lo estructuramos** bien desde un principio luego **es muy difícil continuar.**
- Aprender que **hay veces que es mejor echarlo todo abajo y comenzar de nuevo**, esto no debe preocuparos, debe ser algo habitual. No os vengáis abajo si os pasa, es parte del proceso de aprendizaje. Quiero recordar que hay muchas veces que sabemos

Después de la resolución progresiva del ejercicio os daréis cuenta de que una nueva cultura de desarrollo existe y es la que vamos a asumir de aquí en adelante.

# Un reto por aprender CALIDAD

que algo está mal y continuamos sin mirar atrás pero luego el problema regresa y se hace cada vez mayor. A veces comenzar de nuevo es la mejor opción.

- **Intentar minimizar cualquier error** que se pueda producir, porque tenemos que entender el coste que supone hacia otros departamentos.
- **Comenzar a trabajar con Git** como repositorio de código fuente en lugar de TFS, os animo a que os deis de alta en Github.com con una cuenta gratuita y subáis ahí vuestro proyecto a un repositorio público, de esta forma podéis ir grabando los cambios cada día y volver atrás si es necesario y si necesitáis ayuda puedo visualizar vuestro código sin problema.
- **Yo estaré siempre para ayudaros**, pero solo cuando veáis que no es posible avanzar, no os daré apenas formación, tan solo os diré donde mirar, se trata de que investiguéis y que lo que aprendáis no lo olvidéis nunca, vosotros vais a llegar a la solución yo solo os guiaré para conseguirlo, quizás os enseñe alguna técnica u os daré una pista para que la estudiéis, pero nunca voy a daros la respuesta clara a un problema.

- No os voy a detallar como debe de funcionar, esto **deberéis descubrirlo vosotros** tan solo os pondré un pequeño ejemplo de uso.
- Dado que tenemos en marcha muchos proyectos ordinarios del día a día, mi acercamiento a vuestras soluciones se irá desencadenando durante los días siguientes a su lanzamiento de un modo constructivo y colaborativo.
- A medida que vayáis finalizando una fase, os enviare la siguiente, para ello es imprescindible que demostréis que los valores que genera la aplicación son correctos, normalmente os daré unos valores de configuración y os pediré la fecha y descripción de la salida de vuestro programa para realizar las comprobaciones.

***Se trata también de comenzar a conocernos, de colaborar en un proyecto común basado en la calidad de un software más fácil, más eficiente y que nos haga crecer como profesionales.***

Espero y os animo a intentarlo, para algunos no será nada fácil, pero descubriréis una nueva forma de trabajar que os permitirá mejorar como profesionales, os pido también un poco de fe, se que va a conllevar un esfuerzo extra, pero os aseguro que merecerá la pena.

Debéis dar la posibilidad de probar que funciona según las especificaciones establecidas para que **cualquiera pueda testarlo**

- Para este ejemplo no vamos a usar ninguna infraestructura de la compañía, aunque si queréis usar vuestro equipo no hay ningún problema.
- Para ello vamos a dar una pequeña formación introductoria para que podáis usar Git, podéis comenzar a crearos una cuenta aquí [GitHub](#), a través de alguna cuenta de correo personal que usareis a lo largo del proyecto, tener en cuenta que un repositorio de código fuente es imprescindible sobre todo si queréis dar marcha atrás o ver diseños anteriores. En un futuro a corto plazo todas las soluciones de Semicrol va a usar este sistema, así que lo que aprendáis nos servirá a partir de ahora.
- Vamos a enseñaros también como interaccionar con otros compañeros usando Visual Studio Live Share, un complemento muy usado en programación por parejas, disponible tanto en Visual Studio como en [Visual Studio Code](#) como su herramienta de desarrollo. Para esto también podéis usar Teams, lo dejo a vuestra elección.

## Scheduler. *Parte 1 – Complejidad baja*

---

Esta es la primera parte del ejercicio, su complejidad es relativamente baja, sin embargo, es aquí es donde vais a establecer las bases de una buena estructura para que posteriormente pueda escalar sin problemas.

**El cliente no ha especificado en que plataforma va a funcionar, puede que funcione en Web, en Windows, en un dispositivo móvil o que simplemente sea un servicio webservice, rest, azure function, etc, por lo que vuestro desarrollo debe ser agnóstico de la plataforma.** Los formularios que adjunto son únicamente para que tengáis una idea sobre el funcionamiento del proceso, vosotros decidís como resolverlo de la mejor forma.

Se trata de un pequeño programa que parte de una fecha determinada, a través del campo denominado **current date** ubicado en la sección **(Input)**. Mostrado el gráfico 1.

La salida del programa **(Output)** ofrece dos valores, primero **la siguiente fecha de ejecución** seguido de **una pequeña descripción** del proceso.

# Un reto por aprender CALIDAD

7

Como se puede apreciar el cálculo se realiza a partir de la configuración diaria, que puede ser de dos formas, una sola vez (Once) o recurrente (Varias veces).

The 'Schedule' dialog box is shown with the following settings:

- Input:** Current date is 04/01/2020. Calculate next date button is visible.
- Configuration:** Type is 'Once' (dropdown), 'Enabled' checkbox is checked. DateTime is 08/01/2020 14:00. Occurs is 'Daily' (dropdown), 'Every' checkbox is checked, and the interval is 0 day(s).
- Limits:** Start date is 01/01/2020. End date is empty.
- Output:** Next execution time is 08/01/2020 14:00. Description is 'Occurs once. Schedule will be used on 08/01/2020 at 14:00 starting on 01/01/2020'.

The 'Schedule' dialog box is shown with the following settings:

- Input:** Current date is 04/01/2020. Calculate next date button is visible.
- Configuration:** Type is 'Recurring' (dropdown), 'Enabled' checkbox is checked. DateTime is empty. Occurs is 'Daily' (dropdown), 'Every' checkbox is checked, and the interval is 1 day(s).
- Limits:** Start date is 01/01/2020. End date is empty.
- Output:** Next execution time is 05/01/2020. Description is 'Occurs every day. Schedule will be used on 08/01/2020 at 14:00 starting on 01/01/2020'.



- En el primer caso (**Once**) la salida seria:

08/01/2020 14:00

- El segundo caso (**Recurring**) la salida seria:

05/01/2020 0:00

Posteriormente debería ser algo como:

06/01/2020 0:00

07/01/2020 0:00

08/01/2020 0:00

...



## Un reto por aprender CALIDAD

Os aconsejo también llevar el control del tiempo invertido. Se trata de conocer los tiempos que vais empleando para resolverlo, esto es algo únicamente para vosotros, os servirá para evaluar el coste de un diseño y que posteriormente lo podáis comparar, esta información es solo vuestra, yo nunca os la pediré, me parece interesante que hagáis este ejercicio para valorar el coste de cualquier proyecto.

**Puesto que el objetivo es convenceros y podáis evaluar claramente cuál es la mejor alternativa.**

**Podéis usar Excel o una simple tabla en Word de la manera siguiente:**

**Una vez finalizada esta parte y contrastemos que funciona adecuadamente, pasaremos a la segunda fase.**

**Cuando queráis estoy a vuestra disposición.**

Happy Coding!!

Tarea	Tiempo invertido	Observaciones
Diseño del programa	60 minutos	
Diseño del programa	60 minutos	
Pruebas	30 minutos	
Resolver error 1	15 minutos	
Nueva funcionalidad	10 minutos	
....		