**Universidad Autónoma de Yucatán**


**Facultad de Matemáticas**

Construcción de software

5to semestre, grupo B

**Integrantes:**

Russel Bonilla Pech

Jaqueline Góngora Tun

Frida Pineda Alvarado

Víctor Rosado Koyoc

Luisa Villanueva Díaz

**Profesor:** Edwin León

# Checklist de prácticas de construcción

| Lineamiento de la asignatura | Cumple (Sí / No) |
|---|---|
| ***Técnicas de construcción de variables y tipos de datos fundamentales*** | |
| Inicializar cada variable cuando se declara. | |
| Idealmente, declarar y definir cada variable cerca de donde se usa. | |
| Inicializar los datos de los miembros de una clase en su constructor. | |
| Los nombres de las variables deben usar camelCase. | |
| Usar calificadores de cálculos en nombres de variables. | |
| Usar opuestos comunes en nombre de variables.  Usar los opuestos con precisión. | |
| Nombramiento de índices de bucle. Si tiene que usar i, j y k, no usar para otra cosa que no sean índices de bucle. | |
| Si tiene varios bucles anidados, asignar nombres más largos a las variables de bucle para mejorar la legibilidad. | |
| Pensar en un nombre mejor que “flag” para las variables de estado. Una bandera nunca debe tener “flag” en su nombre. | |
| Utilizar nombres descriptivos para variables temporales en lugar de llamarlas como “temp” o “x”. | |
| Tener en cuenta los nombres booleanos típicos (done, error, found, success). | |
| Asignar nombres de variables booleanas que impliquen Verdadero o Falso. | |
| Utilizar nombres de variables booleanos positivos. | |

| | |
|---|---|
| Identificar constantes nombradas. Usar todas las letras mayúsculas, posiblemente con guiones bajos para separar las palabras, RECSMAX o RECS_ MAX. | |
| Diferenciar entre nombres de variables y nombres de rutinas. | |
| Identificar parámetros de solo entrada (i*nput-only*). | |
| Hacer que las conversiones de tipo sean obvias. | |
| Revisar el desbordamiento de enteros. | |
| Revisar si hay desbordamiento en resultados intermedios. | |
| Evitar cadenas y caracteres mágicos | |
| Estar atento a los errores off-by-one | |
| ***Técnicas de organización de sentencias*** | |
| Organizar el código para que las dependencias sean obvias. | |
| Nombrar las rutinas para que las dependencias sean obvias. | |
| Usar parámetros de rutina para hacer que las dependencias sean obvias. | |
| ***Técnicas de construcción de estructuras de control de flujo*** | |
| Escribir primero la ruta nominal a través del código; luego escribir los casos inusuales. | |
| Asegurarse de bifurcar correctamente en igualdad. | |
| Seguir la cláusula if con una declaración significativa. | |
| Poner los casos más comunes primero. If-then-else. | |
| Ordenar casos por frecuencia en las declaraciones case | |

| ***Técnicas de construcción de procedimientos*** | |
|---|---|
| Para nombrar una función, usar una descripción del valor devuelto. Por ejemplo, cos(), customerId.Next(), printer.IsReady0 y pen.CurrentColor(). | |
| Para nombrar un procedimiento, utilizar un verbo fuerte seguido de un objeto. Por ejemplo: PrintDocument(), CalcMonthlyRevenues(), CheckOrderInfo() y RepaginateDocument). | |
| Poner las variables de estado o error al final en las rutinas. | |
| No utilizar parámetros de rutina como variables de trabajo. | |
| Usar parámetros con nombres significativos. | |
| ***Técnicas de construcción de clases*** | |
| Los nombres de las clases deben escribirse con PascalCase y deben ser sustantivos en singular. | |
| Los nombres de las interfaces deben escribirse con PascalCase. | |
| Preferir la herencia a la verificación extensiva de tipos. | |
| Si varias clases comparten un comportamiento común, pero no datos, derivarlas de una clase base común que defina las rutinas comunes. | |
| Si varias clases comparten datos y comportamientos comunes, heredar de una clase base común que define las rutinas y los datos comunes. | |
| Heredar cuando desea que la clase base controle la interfaz; contener cuando desea controlar su interfaz. | |
| Hacer cumplir la propiedad singleton mediante el uso de un constructor privado. | |
| ***Técnicas de documentación de código*** | |

| | |
|---|---|
| Usar espacios en blanco para separar elementos de una sentencia no muy relacionados o mostrar precedencia de operadores. | |
| Uso de la sangría | |
| Escribir comentarios que describan la intención del código. | |
| Hacer que cada comentario cuente. No escribir más comentarios, hacer que el código sea más legible. | |
| Centrar los comentarios de los párrafos en el por qué en lugar del cómo. | |