

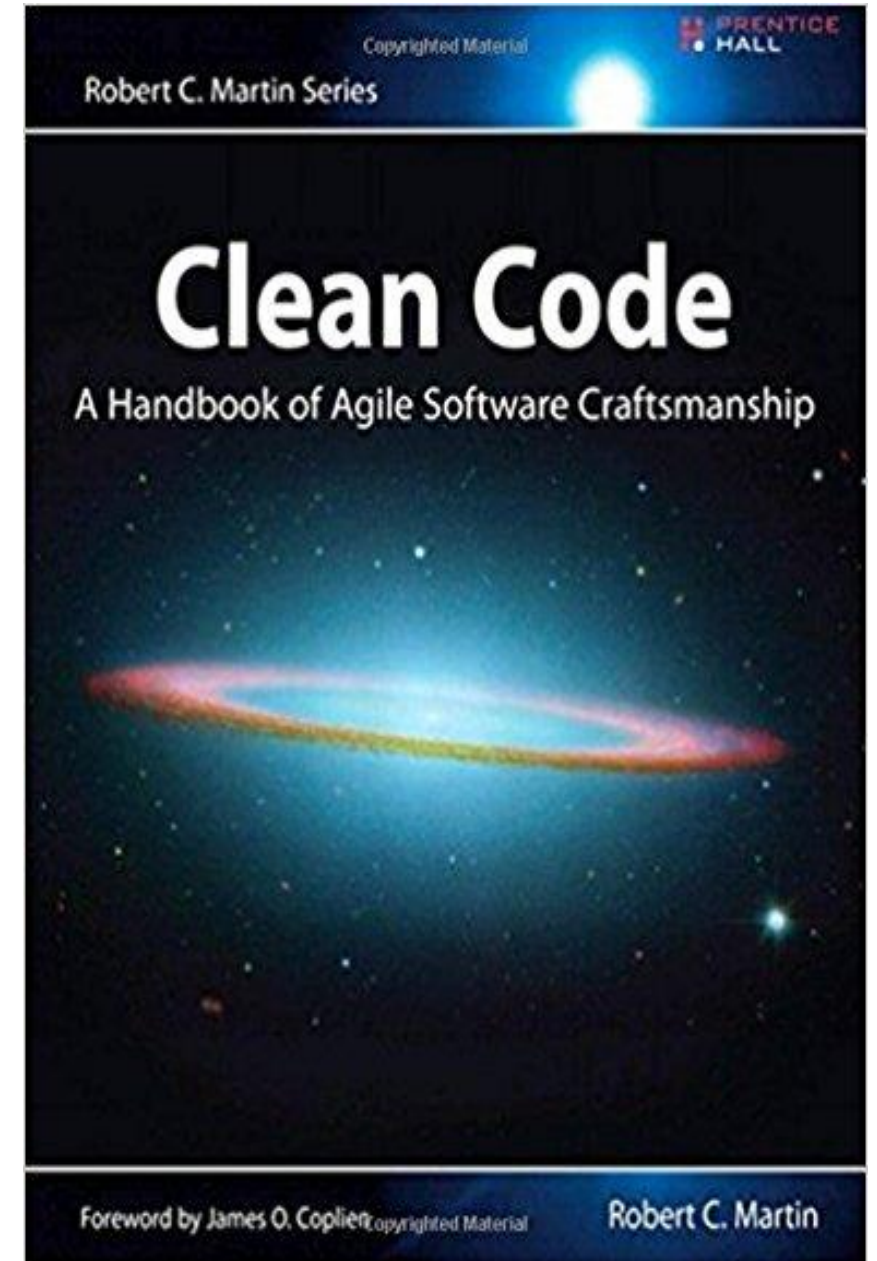
# Código Limpio

Luis Fernando Charpentier Gonzalez

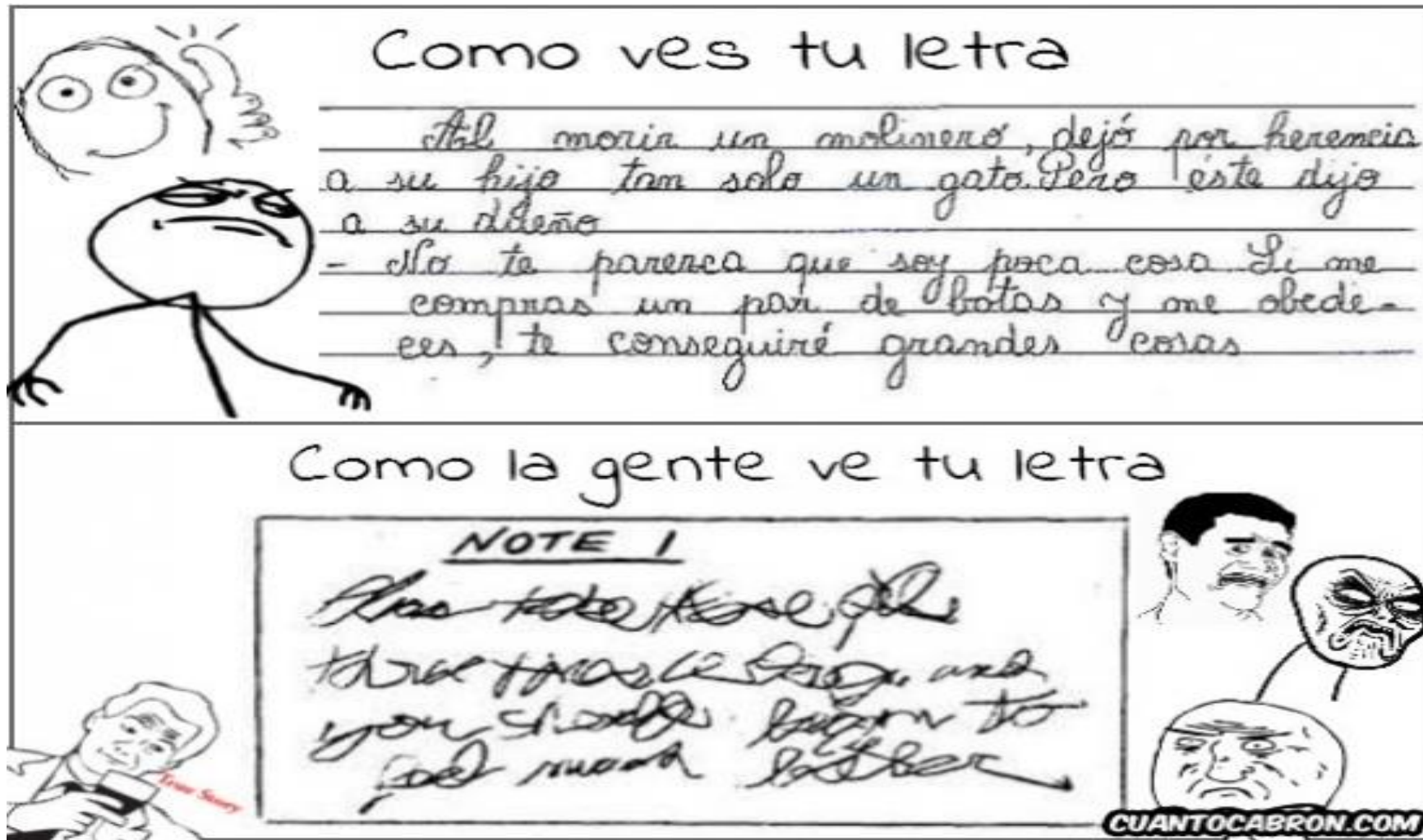


# Clean Code


- El código limpio es un concepto creado por Robert C . Martin en su libro “Clean Code”.



# Caligrafía (arte de escribir con letra bella)



# Caligrafía



Imágenes Educativas.com

Ficha de caligrafía letra r

1. Rosita tiene un ramo
2. Teresa arma la torre
3. El radio es de Renato
4. Sara tiro su ramita
5. Tomás calió a Rusia

<https://www.imageneseducativas.com/>

# Clean

- ¿Alguna vez dejaremos de escribir código fuente?
- ¿Cómo es un Código Limpio?



**Client: Can you fix this bug for me?**



No problem, let me take a look at the code ^.^



**MOTHER OF GOD.**



# Detalles !!!

- Dios esta en los detalles. Arquitecto Ludwin.
- Por que las cosas pequeñas importan ? Pequeños esfuerzos. !!
- Ser profesionales.
  - Organización.
  - Limpieza.
  - Estandarización.
  - Disciplina

“La codificación no se trata solo de decirle a una computadora qué hacer, sino, decirle a la gente lo que tu intentas hacer con una computadora.”



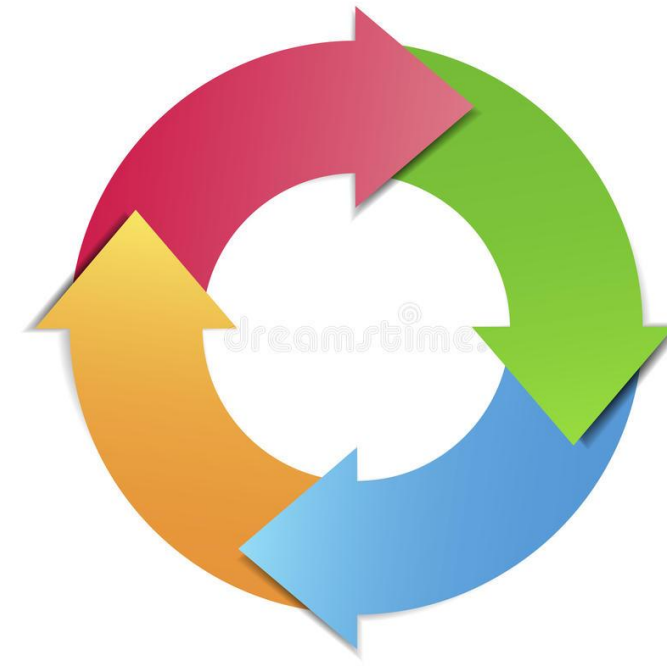


“Cualquier tonto puede escribir código que una computadora puede entender. Los buenos programadores escriben códigos que los humanos pueden entender.”



# Ciclo de vida

“Los desarrolladores vendrán y se irán. Los errores aparecerán. Se agregarán nuevas características. Se pasarán horas y horas trabajando con el mismo código, creciendo y estirándolo”

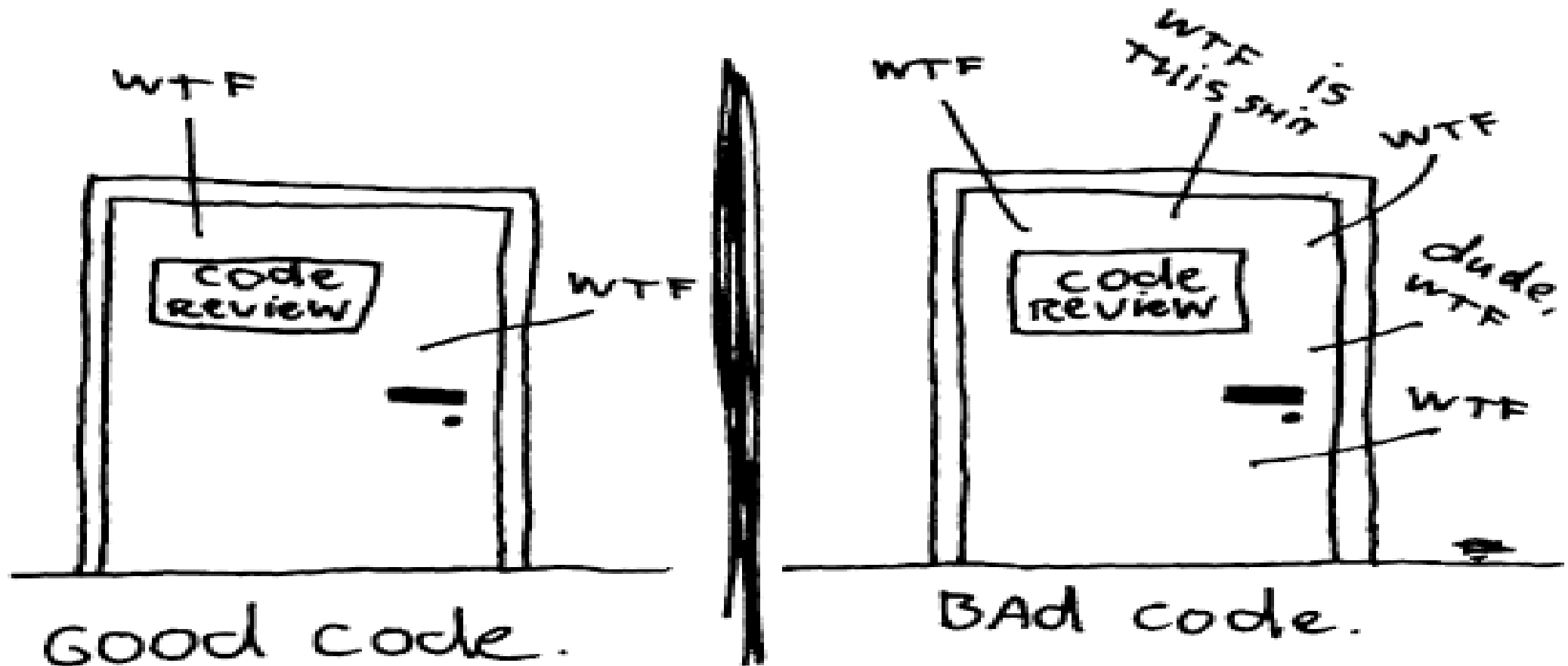


“Si su código es difícil de entender, será más fácil para las personas cometer errores al cambiarlo. ”

“Otros tendrán miedo de eso. No les gustará cambiarlo y hacerlo mejor, por lo que se pudrirá y empeorará con el tiempo. Algo que comenzó como una sola línea complicada comerá su sistema desde adentro hacia afuera, una línea a la vez.”



# The ONLY valid MEASUREMENT OF code QUALITY: WTFs/minute



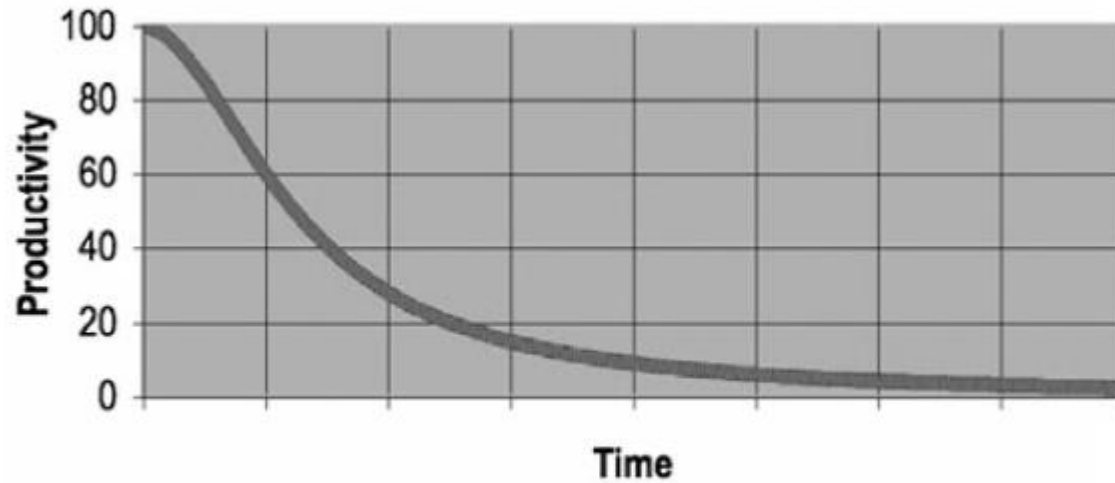


# Consecuencias

- ¿Quién es el responsable del código sucio?
- “Escribo sucio porque voy de prisa.”
- The *only* way to make the deadline—the only way to go fast—is to keep the code as clean as possible at all times.

# Consecuencias

- ¿Cuáles son las consecuencias de mal código?



**Figure 1-1**  
Productivity vs. time

# Somos autores

- Somos autores.
- Y los autores tienen lectores.
- Los autores son los responsables por comunicarse bien con sus lectores.
- La próxima vez que escribamos una línea de código, recordemos que somos los autores que escribimos para lectores, los cuales van a juzgar tu esfuerzo.

# Recordemos

- La regla de los Scouts.





# Fácil de entender

- Escribir código fuente que funciona no es suficiente.
- También debe ser fácil de entender pues en el futuro otros colegas lo van a leer y a modificar.

# Fácil de entender

- Escribir código fuente que otros programadores puedan entender no es una tarea fácil.
- Pero es lo que esperamos de un programador Profesional.

# Cuantos de ustedes se han atrasado por código malo ?

- Con Errores ?
- Desorganizado.
- Nombres malos.
- Difícil de entender.



# Los dos riesgos de un código fuente difícil de entender

“No lo entiendo, necesito más tiempo”





# Los dos riesgos de un código fuente difícil de entender

“Lo entendí mal, e introduje un defecto”



# Nombres significativos

## Meaningful Names

by Tim Ottinger



# La idea general

- Si el código fuente se lee como un texto bien escrito, entonces va a ser más sencillo de entender.
- Si es más sencillo de entender, entonces habrá menos riesgos de defectos.
- Si hubiera un defecto, entonces será más sencillo de encontrar y corregir por una persona que no es el autor del código original.

# Nombres pronunciables

Si usamos palabras completas, otras personas podrán leerlos más fácilmente.

Algunos contra ejemplos:

- `Catch` `dsdpe` `As` `DosSinDatosPreviosException`
- `paramLongLlave`
- `srvParams`



# Ejemplos

**Escribamos lo siguiente en código pronunciable:**

Si la persona es ciudadano de oro, entonces aplique descuento, sino no aplique descuento.

```
if laPersona.EsCiudadanoDeOro  
    AplicarElDescuento();
```

# Ejemplos

**Escribamos lo siguiente en código pronunciable:**

Obtenga el estudiante con el promedio de notas mas alto, luego genere el reporte de notas.

```
elEstudiante = ObtenerElEstudianteConLaNotaMasAlta();  
GenerarElReporteDeNotas(elEstudiante);
```

# Ejemplos

**Escribamos lo siguiente en código pronunciado:**

Si el cliente está moroso, entonces el monto a pagar es igual al monto más 500 colones por día de morosidad.

```
if elCliente.EsMoroso
    elMontoaApagar = monto + (500 * elCliente.CantidadDeDiasDeMorosidad)
```

# Frases *clean*...

- “los compiladores van a optimizar los ciclos y bytes extras que invirtamos para hacer el código mas fácil de entender. Hoy sabemos que el costo del software está en su mantenimiento.”
- “Cada vez que tenemos que ir a leer un código para entender un nombre, entonces sabemos que el código falló en expresar su intención. ¡Es un nombre malo!”

# Nombres

- Los nombres de una variable función o clase deben responder a una serie de cuestiones básicas
  - Por que existe.
  - Que hace.
  - Como se usa.

# Evitemos desinformar

- “Los nombres son nuestras herramientas de comunicación. Entonces debemos pensarlos para que otras personas puedan usarlos para comunicarse.”
- Este método se usa así
  - ObtengaLosMeses (`False`)
- ¿Obtiene los nombres, los códigos, los números?
- ¿Cómo podríamos aclararlo?
- `int d; //Tiempo transcurrido en dias`



# The Telephone Test

A useful way to decide if some piece of code is clear or not is the “telephone test.” If someone could understand your code when read aloud over the telephone, it’s clear enough. If not, then it needs rewriting.

---

*Use the “telephone test” for readability.*

---

The Elements of Programming Style (1978)

# ¿Cómo ponerle nombres a una **variable**?

- Debe especificar que contiene
- Y en que unidad.
  - `int` losDiasDeAtraso.
  - `Int` laDiferenciaDeFechasEnDias
  - `int` lasHorasLaboradas

# ¿Cómo ponerle nombres a Clases?

- Un buen nombre indica lo que la clase es.
- Son sustantivos, como:
  - Estudiante
  - Cuenta
  - Garantía
  - Sesión
  - Cupón

# ¿Cómo ponerle nombres a Métodos?

- Un buen nombre indica lo que el método hace.
  - Son Ordenes o Preguntas.
    - ProceseLaLiquidación
    - GenereLaOrdenDePago
    - IntenteIniciarUnaSesion
    - EstaVencida
    - YaVencio
    - EsCapitalizable
    - ExisteElEstudiante
    - ObtengaElPromedioDeNotas.
    - ObtengaLaNotaMasAlta.
    - ObtengaLaNotaMasBaja.

# Las constantes

- Nos ayudan a poner un nombre a conceptos que no se entienden sólo por su valor.
- El siguiente código podría des actualizarse si tenemos números que son explicados con un comentario.

## ContraEjemplo

```
if (elEmpleado.DiasTrabajados>9)  
    elEmpleado.Gestionar();
```

## Ejemplo

```
const int CantidadMinimaDeDiasParaPagar = 9;  
if (elEmpleado.DiasTrabajados>CantidadMinimaDeDiasParaPagar)  
    elEmpleado.GestioneElPago();
```

# ¿Números, constantes o variables?

- Use un número si no hay un concepto detrás (ej, Fórmulas).

`laTasaNeta = laTasaBruta * (1 - laTasaDeImpuesto)`

- Use constantes o enumerados si necesita aclarar un concepto detrás de un número o `String`.

`Const centimetrosPorPulgada = 2.54`

`Centimetro = pulgadas * centimetrosPorPulgada`



# Condicionales

- Los booleanos se entienden bien si buscamos que se lean como oraciones o predicados.
- Por ejemplo, se leen bien en un **IF**.

```
If LaEmisionEsValida Then  
    IncluyeEnLaValoracion()
```

```
If LlegoTarde Then  
    RegistreLaTardia()
```

```
If !ExisteEstudiante Then  
    RegistreElEstudiante()
```

# El tamaño de los nombres

- Si los nombres se declaran cerca de su uso, pueden tener nombres muy cortos. El contexto es lo que nos permite hacerlos concisos.
- Esto es afectado por el tamaño de nuestras funciones y clases.

Clase + Función + Variable

**Clase:** Estudiante

Atributos: Nombre, Apellidos

**Método:** ObTenerElPromedioNotas

**Clase:** LibroDeClase

Atributos: ListaDeEstudiantes

**Método:** ObTenerElEstudianteConPromedioMasAlto

# Conclusión

- **Si** el código fuente se lee como un texto bien escrito, va a ser más sencillo de entender.
- **Si** podemos entenderlo, podemos extenderlo o corregirlo con más confianza **por alguien que no sea el autor original.**
- **Entonces**, los nombres significativos nos ayudan a:
  - Tener menos defectos
  - Ser más eficientes
  - Cumplir nuestros compromisos con el cliente

# Mejore los nombres

- Siempre piense en mejorar los nombres del código donde trabaje.
- Al hacer un mantenimiento, no tema cambiar los nombres para facilitar el trabajo de las personas a futuro.
- “Cuando pida algo prestado, devuélvalo igual o mejor de como lo recibió.”

# El buen uso de los comentarios en el código fuente.

- Explican el “por qué”

```
' Los saldos de los títulos en UDES ya vienen colonizados  
' por lo que lo unico que hay que hacer es sumarlos con los colones.  
If elSaldoEnUDES <> 0D Then  
|   elSaldoEnColones = elSaldoEnColones + elSaldoEnUDES  
End If
```

# El buen uso de los comentarios en el código fuente.

- Explican código confuso

```
$.fn.exists = function () {  
    return this.length !== 0;  
}
```

```
$(".t-grid .t-refresh").css("display", "none");  
$(".t-header").css("text-align", "center");  
$('#CodMoneda').data('tComboBox').select(0);
```



# El buen uso de los comentarios en el código fuente.

- Explican código procedimental (Extenso)

```
' validar si el ISIN existe en RNVI ( 1ra validacion)
Dim valorOficial As Model.EmissionAutorizada = laListaDeEmisiones.FirstOrDefault(Function(item) item
If valorOficial Is Nothing Then
    mBitacora.Add(String.Format(MENSAJE_NO_EXISTE_EN_REGISTRO_OFICIAL, laFecha, activoLocal.CodIsir
    Exit For
End If

' valida si el Emisor existe ( 2da validación)
Dim elEmisor = laListaDeEmisores.FirstOrDefault(Function(item) item.DesEmisor.Equals(valorOficial.[
If elEmisor Is Nothing Then
    mBitacora.Add(String.Format(MENSAJE_EMITOR_NO_REGISTRADO, laFecha, valorOficial.DesEmisor, acti
    Exit For
End If

' validar si la moneda de activo es válida (3ra validacion)
If Not MonedasPermitidas.Contains(valorOficial.CodMoneda) Then
    mBitacora.Add(String.Format(MENSAJE_MONEDA_DE_ACTIVO_NO_ES_VALIDA, laFecha, activoLocal.CodIsir
    Exit For
End If
```

# Mal uso de los comentarios en el código fuente.

- Comentarios que no aportan (distracción)

```
''' <summary>  
''' Valida los datos del item.  
''' </summary>  
''' <param name="item">Item a validar.</param>  
''' <returns>True.</returns>
```

31 references

```
Public Overrides Function Validar(item As Transferencia) As Boolean
```

# Mal uso de los comentarios en el código fuente.

- Comentarios de fechas y responsables

```
/**
 * Nombre: ██████████
 *
 * Cliente: BCCR
 *
 * Autor: ██████████
 *
 * Fecha: Enero 2013
 *
 * Descripción:
 * Esta clase se encarga de funcionar como un despachador de mensajes swift
 *
 * Última modificación hecha por:
 *


| Nombre     | Fecha      | Descripción               |
|------------|------------|---------------------------|
| ██████████ | 14/01/2013 | Creación Inicial de clase |


 */
public class BCCR MonitorMensajesSWIFT implements IScript {
```

# Mal uso de los comentarios en el código fuente.

- Comentarios mentirosos

```
' Si el item no existe, no hace nada adicional y se retorna True.  
If IsNothing(pagoPorAbortar) Then  
    Seguimiento.Escribir("Se notificará del aborto de algo inexistente")  
    EspecificacionNotificar.DispararEventoAbortoTransferenciaNoExiste(fe  
  
    'Se registra en bitácora que se ignoró  
    accionesTransferencia.EscribirMensajeBitacoraAbortoPagoInexistente(r  
    Return False  
End If
```

---

# Mal uso de los comentarios en el código fuente.

- Código comentado

```
'TODO: Este código se comenta, ya que el monto en UDES viene colonizado.  
  
' -- Logica cuando el nominal de los UDES estaba en UDES.  
'Dim montoTotalUDES As Decimal  
' Los saldos en UDES se convierten de acuerdo al tipo de cambio actual  
'Dim procesoConversor As New Acciones.TipoCambioUdes  
'montoTotalUDES = procesoConversor.ConvertirUdesEnColones(saldoUDES.MtoSaldoAportado)
```

# Mal uso de los comentarios en el código fuente.

- Dudas

```
If cobertura Is Nothing Then
    mBitacora.Add(String.Format(MENSAJE_PLAZO_FUERA_DEL_RANGO_DE_COBERTURAS,
                                laFecha,
                                activoLocal.
                                CodIsin,
                                valorOficial.FecVencimiento,
                                diasAlVencimiento,
                                activoLocal.CodEntidad))

    ' TODO: Validar esta logica con el negocio
    ' Si no encontrara la cobertura envía un mensaje a bitacora y lo intenta valorar con porcentaje de cobertura "0"
    cobertura = New Model.GradoCoberturaPorTipoActivoYPlazo With {.PorcentajeCobertura = 0}
    detalleActivo = Me.GenerarValoracionPorISIN(laFecha,
                                                elEmisor,
                                                activo,
                                                valorOficial,
                                                cobertura,
                                                vectorPrecio)

    valoracionesPorISIN.Add(detalleActivo)
Exit For
End If
```



# Mal uso de los comentarios en el código fuente.

- Dudas

```
If cobertura Is Nothing Then
    mBitacora.Add(String.Format(MENSAJE_PLAZO_FUERA_DEL_RANGO_DE_COBERTURAS,
                                laFecha,
                                activoLocal.
                                CodIsin,
                                valorOficial.FecVencimiento,
                                diasAlVencimiento,
                                activoLocal.CodEntidad))

    ' TODO: Validar esta logica con el negocio
    ' Si no encontrara la cobertura envía un mensaje a bitacora y lo intenta valorar con porcentaje de cobertura "0"
    cobertura = New Model.GradoCoberturaPorTipoActivoYPlazo With {.PorcentajeCobertura = 0}
    detalleActivo = Me.GenerarValoracionPorISIN(laFecha,
                                                elEmisor,
                                                activo,
                                                valorOficial,
                                                cobertura,
                                                vectorPrecio)

    valoracionesPorISIN.Add(detalleActivo)
Exit For
End If
```

# Mal uso de los comentarios en el código fuente.

- Cuando dicen lo que ya leemos en el código

```
'Se suman los UDES y los colones, solo si los UDES no son cero  
If elSaldoEnUDES <> 0D Then  
|   elSaldoEnColones = elSaldoEnColones + elSaldoEnUDES  
End If
```

# Mal uso de los comentarios en el código fuente.

- No se entienden

```
<TestMethod()> _  
Public Sub ValorarUnActivo_ExisteEnRNVI_EmisorInexistente()  
    ' TODO: Corregir  
    Dim emisores2 As New List(Of Model.Emisor)  
    Dim ISIN = "CRBCCR0B3207"  
    Dim activosDelFideicomiso As New List(Of Model.SaldoAportadoDeta  
    Dim unActivoDelFideicomiso = Generador.NuevoActivo(fechaDeProces  
    activosDelFideicomiso.Add(unActivoDelFideicomiso)  
  
    Dim retorno = bs.GenerarValoracionesPorISIN(fechaDeProceso, acti  
    Assert.IsTrue(bs.MensajesDeBitacora.Contains(String.Format(Accio  
  
End Sub
```

# Mal uso de los comentarios en el código fuente.

- Código sin terminar

```
Catch ex As Exception
    Infraestructura.EscribaElMensajeEnBitacora($"Gsp - Error: {ex.Message}",
                                                ex.ToString)

    elResultado.Resultado = ResultadoEjecucion.ErrorEnEjecucion
    elResultado.Detalle = {"Falló el proceso de importación " &
                          $"de movimientos para la fecha: {laFecha:dd/MM/yyyy} " &
                          $"con el siguiente error: {ex.Message}"}

    'TODO: Notificar
End Try
Return elResultado
End Function
```

```
static void Copiar(char[] a1, char[] a2)
{
    for (int i = 0; i < a1.Length; i++)
    {
        a2[i] = a1[i];
    }
}
```

```
static void Copiar(char []fuente,char []destino)
{
    for (int contador = 0; contador < fuente.Length; contador++)
    {
        destino[contador] = fuente[contador];
    }
}
```

```
class DtaRcr102
{
    private DateTime genymdhms;
    private DateTime modymdhms;
    private string ps7qint = 102;
}
```

```
class Cliente
{
    private DateTime EstampaDeTiempoDeGeneracion;
    private DateTime EstampaDeTiempoDeModificación;
    private string IdentificadorDelRegistro = "102";
}
```



```
for (int j = 0; j < 34; j++)  
    {  
        s += (t[j]*4)/5;  
    }
```

```
const int CantidadDeDiasTrabajadosPorSemana = 5;
const int NumeroDeTareas = 34;

int cantidadDeDiasIdealesParaUnaTarea = 4;
int cantidadTotalDeSemanasATrabajar=0;

for (int j = 0; j < NumeroDeTareas; j++)
{
    int cantidadDeDiasRealesPorTarea = cantidadDeEsfuerzoEstimado[j] *
cantidadDeDiasIdealesParaUnaTarea;
    int cantidadRealDeSemanasPorTarea = (cantidadDeDiasRealesPorTarea
/ CantidadDeDiasTrabajadosPorSemana);
    cantidadTotalDeSemanasATrabajar += cantidadRealDeSemanasPorTarea;
}
```