



# Curso de QA & Software Testing

## Taller Caja Blanca

# Capacitador

**Ronnel Vélez Manzano**

Gerente de QA & SW Testing

Miembro del HASTQB



**Certificaciones:** SCRUM MASTER

ISTQB | CTFL | Certified Tester Foundation Level

ISTQB | CTAL-TM | Certified Tester Advanced Level - Test Manager

ISTQB | CTFL- AT | Certified Agile Tester

**E-mails:** rvelez@crnova.com

rvelez@hastqb.org

ronnel.velez@gmail.com

Servicios Computacionales Novacomp

San Jose – Costa Rica

**Web:** <http://www.crnova.com>

**Teléfono (Costa Rica):** (506) 2216-5800

**Fax:** (506) 2216-5900



**Microsoft Partner**

Gold Cloud Platform  
Silver Cloud Productivity  
Silver Collaboration and Content  
Silver Application Development  
Silver Small Business



# Instrucciones

El presente material de formación ha sido elaborado por Servicios Computacionales Novacomp. Está basado en el programa de estudios de:

- IREB: International Requirements Engineering Board
- ISTQB: International Software Testing Qualifications Board
- Manifesto for Agile Software Development
- Scrum Alliance

Las marcas contenidas e incorporadas en los documentos son propiedad de sus propietarios respectivos incluso si no son mencionados de forma explícita.

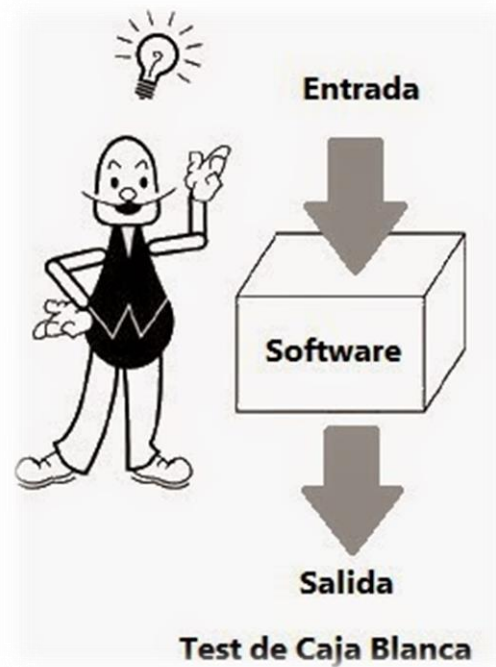


## Microsoft Partner

Gold Cloud Platform  
Silver Cloud Productivity  
Silver Collaboration and Content  
Silver Application Development  
Silver Small Business



# Técnicas basadas en caja blanca (“white box”)



## Técnicas basadas en caja blanca (“white box”)

1.- ¿Para la siguiente porción de código, cuantos casos de prueba son necesarios para alcanzar la cobertura de Decisión del 100%?

```
PrintOutHeader;  
PrintOutSalution;  
PrintOutMainbody;  
PrintOutSignature;
```

# Técnicas basadas en caja blanca (“white box”)

## 2.- Considere la siguiente función:

```
double interest(double avg_balance, double annual_rate)
/* Calcula el interés mensual para una          *
 * cuenta corriente que genera interés, basando el *
 * interés en el saldo promedio mensual.          *
 * Si el saldo medio es menor a cero, es calculado *
 * un interés negativo, pero otros módulos        *
 * tratan los cargos de sobregiro.                  *
 * La tasa anual es expresada como un porcentaje. */
{
    double calc_int=0.0;
    if (avg_balance > 0.0) {
        double monthly_rate = annual_rate/12.0;
        calc_int = avg_balance*(monthly_rate/100.0);
    }
    return(calc_int);
}
```

Suponga que tiene un arnés de prueba que le permitirá presentar los valores de prueba a la función de interés (Las Entradas) y para comprobar los valores que devuelve (Las Salidas), redondeando al céntimo más cercano. ¿Cuál de los siguientes casos de prueba alcanza el 100% de la cobertura de decisión con el número mínimo de pruebas? Suponga que las entradas son los dos primeros números en cada triple y la salida es el tercero

- A. 100.0,5.0,0.42
- B. 100.0,5.0,0.42;-50.0,1.25,0.0
- C. 100.0,5.0,0.42;0.0,2.5,0.0;-50.0,1.25,0.0
- D. 100.0,5.0,0.42;0.01,25.0,0.0;0.0,2.5,0.0;-0.01,10.0,0.0;-50.0,1.25,0.0

## Técnicas basadas en caja blanca (“white box”)

3.- ¿Para la siguiente porción de código, cuantos casos de prueba son necesarios para alcanzar la cobertura de sentencia del 100%?

### Procedure X

```
Read (Color)    //Color de entrada del usuario
IF (Color == “Red”) THEN
    Call Roses(Color)
ELSEIF (Color == “Blue”) THEN
    CALL Violets(Color)
ELSE
    PRINT “User is no Shakespeare”
END IF
SaveToDatabase(Color)
End Procedure X
```

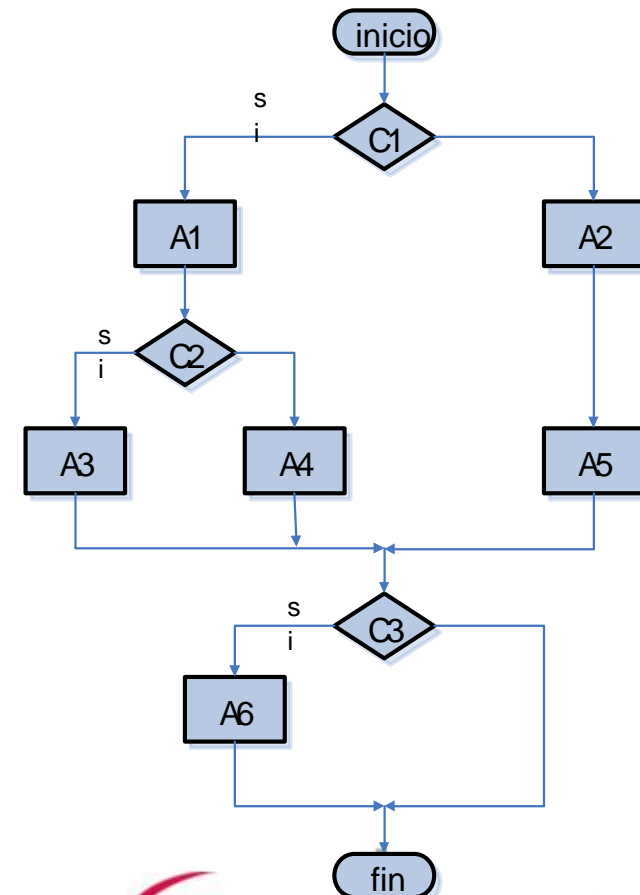
# Técnicas basadas en caja blanca (“white box”)

4.- Crear el mínimo número de casos de prueba para alcanzar una cobertura del 100% de:

- ❑ Decisión o Ramas (“branches”)
- ❑ Condición (multiple) o Camino (path) según aplique

Escribir los casos de prueba como una cadena de condiciones e instrucciones

(Por ejemplo: Inicio, C1, A2, Fin)





## Técnicas basadas en caja blanca (“white box”)

5.- Determinar cuantos casos de prueba mínimo son necesarios para obtener el 100% de cobertura de sentencia.

```
1  int foo (int a, int b, int c, int d, float e) {  
2      float e;  
3      if (a == 0) {  
4          return 0;  
5      }  
6      int x = 0;  
7      if ((a==b) OR ((c == d) AND bug(a) )) {  
8          x=1;  
9      }  
10     e = 1/x;  
11     return e;  
12 }
```

# Técnicas basadas en caja blanca (“white box”)

6.- Realizar un diagrama de flujo de control con nodos y aristas para el siguiente código. Adicionalmente determinar la cantidad de casos mínimos necesarios para dar el 100% de cobertura de sentencia y 100% de cobertura de condición multiple ó Camino (path) según aplique.

```

Start
if C1 true
    then do P1
        if C2 true
            then do P3
        else do P4
        endif
    else do P2, P5
    endif
if C3 true
    then do P6
else
    if C4 true
        then do P7
        else do P8, P9
        endif
    do P10, P11
endif
End
    
```

