Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño

Equipo: Jareth Izhar Aparicio Lopez
Carlos Alberto Guzman Ramirez
Luis Angel Martinez Zamaniego

“Trabajo en clase excepciones”

grupo 932

Organización de computadoras

Ensenada, Baja California a 25 de Noviembre de 2025.

1. ¿Qué son los Breakpoints en programación?

Un breakpoint (punto de ruptura) es una herramienta dentro de un depurador (debugger) que permite pausar la ejecución del programa en una línea específica.
Sirve para:

- Revisar valores de variables.
- Verificar el flujo de ejecución.
- Detectar errores lógicos.
- Analizar el estado del programa paso por paso.

En pocas palabras: es como poner una “pausa inteligente” al programa para ver qué está pasando por dentro.

2. Breakpoints en VSCode

En Visual Studio Code, se pueden usar breakpoints de varias formas:

Activar un breakpoint:
Abre tu archivo de código.
Haz clic en la parte izquierda del número de línea -> aparecerá un punto rojo.
O presiona F9 sobre una línea.

Ejecutar en modo depuración:
Presiona F5.
Selecciona el entorno (C, C++, Python, Assembly con extensión apropiada, etc.).

Herramientas del debugger:
Step Over (F10): avanza sin entrar en funciones.
Step Into (F11): entra dentro de funciones.
Step Out (Shift+F11): sale de una función.
Variables: muestra valores en tiempo real.
Call Stack: muestra el flujo de llamadas.
Watch: permite monitorear variables específicas.

3. Snippet en ensamblador que simule Try/Catch usando saltos

; Simulación de TRY/CATCH en ensamblador x86

```
section .text
global _start

_start:

TRY_BLOCK:
    ; ----------------------------
    ; Código del "try"
    ; ----------------------------

    mov eax, [valor]      ; cargar un valor
    cmp eax, 0            ; ¿ocurrió un error simulado?
    je ERROR_HANDLER      ; si es 0, saltar al catch

    ; Si no hubo error
    jmp END_TRY

ERROR_HANDLER:
    ; ----------------------------
    ; Código del "catch"
    ; ----------------------------
    mov ebx, 1            ; manejar el error
    jmp END_PROGRAM

END_TRY:
    mov ebx, 0            ; todo salió bien

END_PROGRAM:
    mov eax, 1
    int 0x80
```

4. Programa en C usando setjmp y longjmp:

```c
#include <stdio.h>
#include <setjmp.h>

jmp_buf buffer;

int procesarNumero(int x) {
    if (x < 0) {
        printf("Error: número negativo detectado\n");
        longjmp(buffer, 1);  // Salta al "catch"
    }
    return x * 2;  // Regla de negocio: procesar números positivos
}

int main() {
    int valor = -5;   // valor de prueba

    if (setjmp(buffer) == 0) {
        // TRY
        printf("Procesando número...\n");
        int r = procesarNumero(valor);
        printf("Resultado: %d\n", r);
    } else {
        // CATCH
        printf("Excepción capturada: número inválido\n");
    }

    printf("Programa terminado correctamente\n");
    return 0;
}
```

5. Snippet demostrando un breakpoint:

```c
#include <stdio.h>

int main() {

    int a = 10;      // <-- breakpoint aquí
    int b = 20;
    int suma = a + b; // <-- breakpoint aquí

    printf("La suma es: %d\n", suma);
```

```
    return 0;
}
```