

PRUEBAS DE SOFTWARE

1. Prueba de Caja Blanca: Ruta Básica

Método Analizado: btnGuardar_Click()

Clase: SAG__Diploma.Vista.FormRutinas

A. Código Fuente Analizado

```
private void btnGuardar_Click(object sender, EventArgs e)
{
    try { 1
        using (var context = new DiplomaContext())
        {
            var rutinaDb = context.Rutinas
                .Include(r => r.DiasRutinas)
                .ThenInclude(d => d.EjerciciosAsignados)
                .FirstOrDefault(r => r.IdRutina == _rutina.IdRutina);

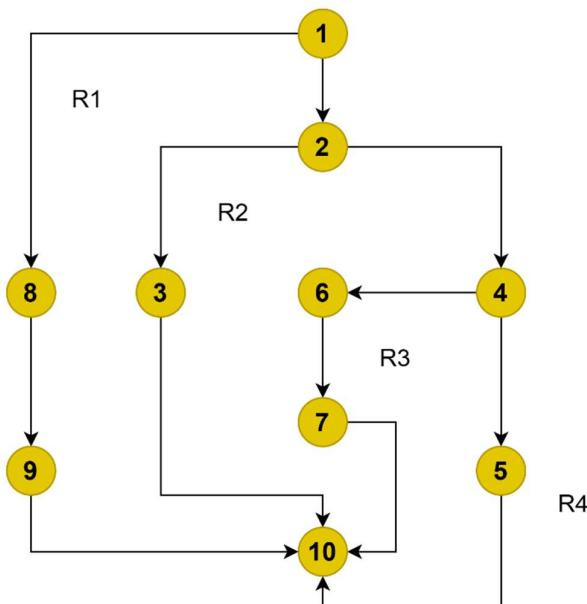
            if (rutinaDb == null) 2
            {
                MessageBox.Show("No se encontró la rutina en la base de datos.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            if (rutinaDb.DiasRutinas.Any(d => d.EjerciciosAsignados.Count == 0)) 4
            {
                MessageBox.Show("Todos los días deben tener al menos un ejercicio asignado antes de guardar.", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            rutinaDb.FechaAsignacion = DateOnly.FromDateTime(DateTime.Now); 6
            context.SaveChanges();

            MessageBox.Show("Rutina guardada correctamente.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        this.Close(); 7
    } catch (Exception ex) 8
    {
        MessageBox.Show($"Error al guardar la rutina: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
} 10
```

B. Gráfica de Flujo



C. Complejidad Ciclomática

Fórmula: $V(G) = \text{Nodos predicado} + 1$

$V(G) = 4$ regiones

$V(G) = 3 \text{ Nodos predicado} + 1 = 4$

$V(G) = 12 \text{ aristas} - 10 \text{ nodos} + 2 = 4$

Por lo tanto, existen 4 Rutas Independientes.

D. Rutas Independientes y Casos de Pruebas

Ruta 1: 1-8-9-10

- **Valores de entrada:** Click en btnGuardar, simulando una falla de infraestructura.
- **Valores esperados:** El sistema captura el error y muestra el mensaje: "Error al guardar la rutina: [Detalle de la excepción]".

Ruta 2: 1-2-3-10

- **Valores de entrada:** Click en btnGuardar luego de que el registro de la rutina haya sido eliminado directamente de la base de datos por un factor externo (para forzar que rutinaDb == null).
- **Valores esperados:** El sistema muestra el mensaje: "No se encontró la rutina en la base de datos." y cancela la operación.

Ruta 3: 1-2-4-6-7-10

- **Valores de entrada:** Click en btnGuardar teniendo una rutina válida cargada, donde todos los días (DíasRutinas) contienen al menos un ejercicio asignado.
- **Valores esperados:** El sistema actualiza la fecha, guarda los cambios en la BD, muestra el mensaje "Rutina guardada correctamente." y cierra el formulario.

Ruta 4: 1-2-4-5-10

- **Valores de entrada:** Click en btnGuardar teniendo una rutina en la que intencionalmente se ha dejado al menos uno de sus días sin ningún ejercicio asignado (EjerciciosAsignados.Count == 0).
- **Valores esperados:** El sistema muestra el mensaje de advertencia: "Todos los días deben tener al menos un ejercicio asignado antes de guardar." y cancela la operación de guardado.

2. Prueba de Caja Negra: Partición Equivalente

Consignas

En base al caso de uso "Guardar Rutina", derivar casos de prueba por el método de Partición Equivalente. Según la condición de entrada, definir las clases de equivalencia válidas e inválidas para cada entrada del dominio.

Dominio de Entrada

- Cantidad de ejercicios asignados por día.
- Estado de la rutina en el sistema.

Atributo: Cantidad de ejercicios por día

- **Tipo condición:** Rango Numérico
- **Entrada Válida:** Todos los días de la rutina tienen 1 o más ejercicios asignados.
- **Entradas Inválidas:** Al menos un día de la rutina no tiene ejercicios asignados.
- **Resultados:** Si la entrada es válida, el sistema permite la persistencia de los datos. Si es inválida, el sistema frena la operación y advierte al usuario.

Atributo: Estado de la Rutina

- **Tipo condición:** Existencial / Booleana
- **Entrada Válida:** El identificador de la rutina existe actualmente en la Base de Datos.
- **Entradas Inválidas:** El identificador de la rutina no existe
- **Resultados:** Si es válido, se procede a validar los ejercicios. Si es inválido, se informa el error y se cancela el guardado.

Casos de Prueba

Caso	Entrada de datos	Resultados de salida
Caso 1:	Se intenta guardar una rutina donde el Día 1 tiene 3 ejercicios asignados, pero el Día 2 no tiene ejercicios (lista vacía).	El sistema muestra un mensaje emergente " <i>Todos los días deben tener al menos un ejercicio asignado antes de guardar</i> " cancelando la operación.

Caso	Entrada de datos	Resultados de salida
Caso 2:	Se intenta guardar una rutina completa, donde todos los días creados (Día 1, Día 2, etc.) tienen al menos 1 ejercicio asignado correctamente.	El sistema muestra un mensaje emergente " <i>Rutina guardada correctamente</i> " continuando con el cierre de la ventana y finalizando la operación de forma exitosa.
Caso 3:	Se intenta guardar una rutina con ejercicios válidos, pero la rutina ya no existe en la base de datos (fue eliminada desde otra terminal antes de presionar Guardar).	El sistema muestra un mensaje emergente " <i>No se encontró la rutina en la base de datos</i> " cancelando la operación para evitar inconsistencias.

Resultados de la Prueba

Los resultados obtenidos en la ejecución de los casos de prueba por partición equivalente coinciden con los resultados esperados. El sistema controla correctamente los límites y reglas del negocio, por ende, **la prueba ha sido exitosa**. No hay que realizar correcciones.