

Prueba de Caja Blanca

“Gestión de información académica InClass”

Integrantes:

Stephen Drouet
Bryan Morales
Alejandro Sarmiento
Jairo Quilumbaquin

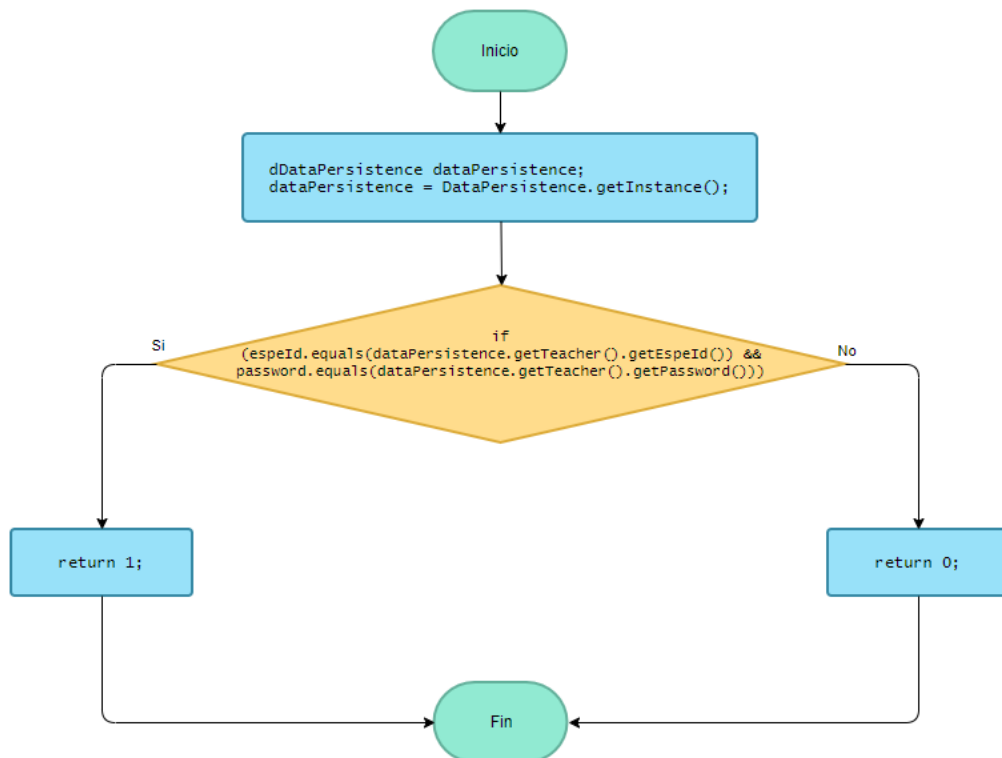
Fecha 21/02/2024

Prueba caja blanca login

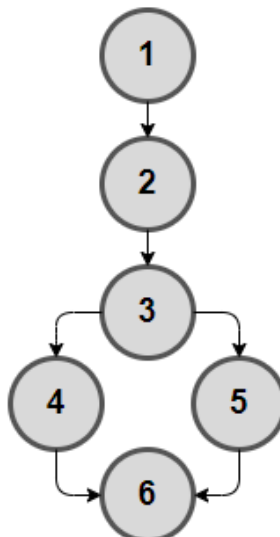
CÓDIGO FUENTE

```
public static int loginTeacher(String espeId, String password) {  
    DataPersistence dataPersistence;  
    dataPersistence = DataPersistence.getInstance();  
    if (espeId.equals(dataPersistence.getTeacher().getEspeId()) &&  
        password.equals(dataPersistence.getTeacher().getPassword())) {  
        return 1;  
    } else {  
        return 0;  
    }  
}
```

DIAGRAMA DE FLUJO



Grafo



RUTAS

R1: 1, 2, 3, 4, 6

R2: 1, 2, 3, 5, 6

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = 1 + 1 = 2$
- $V(G) = A - N + 2$
 $V(G) = 6 - 6 + 2 = 2$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

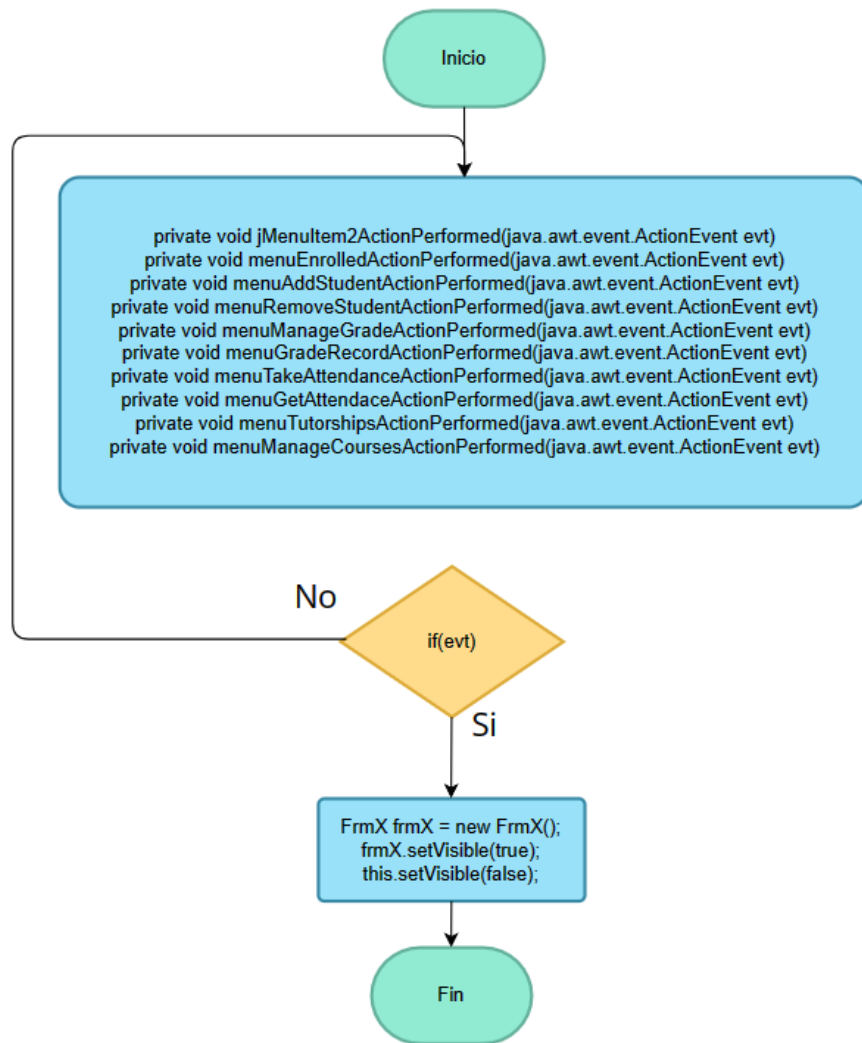
Prueba caja blanca navegar menu

CÓDIGO FUENTE

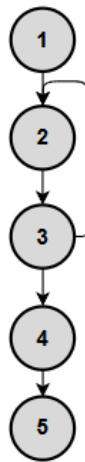
```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)
private void menuEnrolledActionPerformed(java.awt.event.ActionEvent evt)
private void menuAddStudentActionPerformed(java.awt.event.ActionEvent evt)
private void menuRemoveStudentActionPerformed(java.awt.event.ActionEvent
evt)
private void menuManageGradeActionPerformed(java.awt.event.ActionEvent
evt)
private void menuGradeRecordActionPerformed(java.awt.event.ActionEvent
evt)
private void menuTakeAttendanceActionPerformed(java.awt.event.ActionEvent
evt)
private void menuGetAttendaceActionPerformed(java.awt.event.ActionEvent
evt)
private void menuTutorshipsActionPerformed(java.awt.event.ActionEvent evt)
private void menuManageCoursesActionPerformed(java.awt.event.ActionEvent
evt)

if (evt) {
FrmX frmX = new FrmX();
frmX.setVisible(true);
this.setVisible(false);
}
```

DIAGRAMA DE FLUJO



GRAFO



RUTAS

R1: 1, 2, 3, 4, 5

R2: 1, 2, 3, 2, 3, 4, 5

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$
 $V(G) = 5 - 5 + 2 = 2$

DONDE:

P: Número de nodos predichado

A: Número de aristas

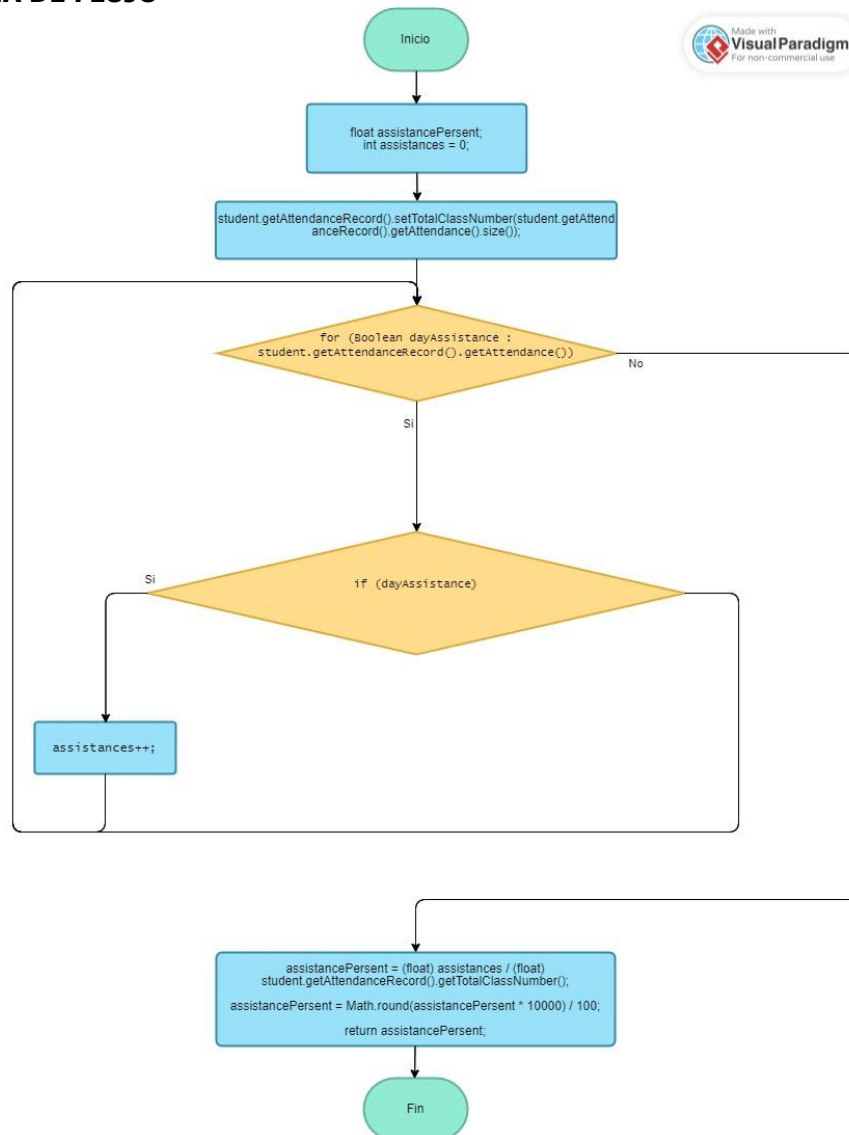
N: Número de nodos

Prueba caja blanca calcular porcentaje de asistencias

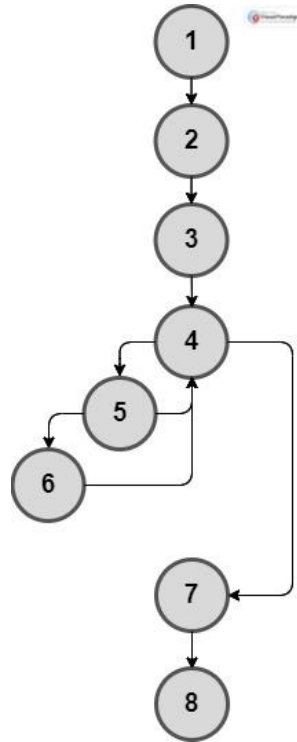
CÓDIGO FUENTE

```
public static float calculateAssistancePersetn(Student student) {  
    float assistancePerset;  
    int assistances = 0;  
    student.getAttendanceRecord().setTotalClassNumber(student.  
getAttendanceRecord().getAttendance().size());  
    for (Boolean dayAssistance : student.getAttendanceRecord().  
getAttendance()) {  
        if (dayAssistance) {  
            assistances++;  
        }  
    }  
  
    assistancePerset = (float) assistances / (float)  
student.getAttendanceRecord().getTotalClassNumber();  
  
    assistancePerset = Math.round(assistancePerset * 10000) / 100;  
  
    return assistancePerset;  
}
```

DIAGRAMA DE FLUJO



GRAFO



RUTAS

R1: 1, 2, 3, 4, 7, 8

R2: 1, 2, 3, 4, 5, 4, 7, 8

R3: 1, 2, 3, 4, 5, 6, 4, 7, 8

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$
 $V(G) = 9 - 8 + 2 = 3$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos