**PRUEBAS UNITARIAS DEL SISTEMA “PROFORMAS-DISTRIBUIDORAJR”**

Las pruebas que se presentaran a continuación están realizadas sobre cada uno de los componentes que forman parte de la aplicación.

1. **Prueba de Fecha Actual.**

@Test

public void FechaActual() {

frmOfertas fecha = new frmOfertas();

Assert.assertEquals("2018/02/22", fecha.fechaActual());

}

1. **Prueba de fechas validas mayores.**

@Test

public void FechasMayoresValidasEnOfertas(){

frmOfertas fecha = new frmOfertas();

String comparar=fecha.compararFechasConDate("2018/02/24", "2018/02/22");

Assert.assertEquals("MA",comparar);

}

1. **Prueba de fechas validas iguales.**

@Test

public void FechasIgualesValidasEnOfertas(){

frmOfertas fecha = new frmOfertas();

String comparar=fecha.compararFechasConDate("2018/02/22", "2018/02/22");

Assert.assertEquals("IG",comparar);

}

1. **Prueba validación de ID.**

@Test

public void ValidacionID() {

frmregistrocliente id = new frmregistrocliente();

assertTrue(id.Verificar("1723764021"));

}

1. **Prueba de validación de ofertas**

@Test

public void AplicacionOfertas() {

OfertaDAO Oferta = new OfertaDAO();

//CODIGO, valor de oferta, metodo

//104.106.303

assertEquals("303", 40, Oferta.OfertaxP("303"), .0);

}

1. **Generación de código.**

@Test

public void GeneracionCodigo() {

String codigo; String[] cod;int numero;

ProformaDAO proformaDAO = new ProformaDAO();

codigo = proformaDAO.GenerarCodigo();

cod = codigo.split("JR000");

numero = Integer.parseInt(cod[1]);

assertEquals("2", 2, numero, 0);

}

1. **Prueba Comprobación de Registro vendedor**

@Test

public void ValidacionExistenciaVendedor() {

VendedorDAO vendedorDAO = new VendedorDAO();

VendedorVO vendedorVO = new VendedorVO("eve","1234", "1723764021", "Anagumbla","Evelyn", "022384010", "QUITO", "1996-07-05");

//Si el vendedor ya esta registradotiene que dar falso

assertFalse(vendedorDAO.registrarVendedor(vendedorVO));

}

1. **Prueba Comprobación de Registro cliente**

@Test

public void ValidacionExistenciaCliente() {

ClienteDAO clienteDAO = new ClienteDAO();

ClienteVO clienteVO = new ClienteVO("Pacha", "Maycol", "2384162", "QUITO", "1996-01-03", "1722293048");

assertFalse(clienteDAO.registrarCliente(clienteVO));

}

1. **Prueba Comprobación Registro proformas.**

@Test

public void ValidacionExistenciaProforma() {

ProformaDAO proformaDAO = new ProformaDAO();

ProformaVO proformaVO = new ProformaVO("JR0000","1707115158","1723764021","2018-02-08","2018-02-15","APROBADO");

assertFalse(proformaDAO.registarProforma(proformaVO));

}

1. **Prueba Comprobación existencia de ofertas.**

@Test

public void ValidacionExistenciaOferta() {

OfertaDAO OfertaDAO=new OfertaDAO();

OfertaVO ofertavo=new OfertaVO("Linea de Hogar","40'","2018/02/25","2018/02/28");

//Verifica si ya existe la oferta

assertTrue(OfertaDAO.registrarOferta(ofertavo));

}

**EJECUCION DE LAS PRUEBAS**

