

Casos de pruebas:

. CasoDePrueba1

. Descripción: Insertar un nuevo alumno en la tabla Alumnos.

. Precondiciones: No debe existir ningún registro en la tabla Alumnos con el id_alumno igual a 0001.

Postcondiciones: Se debe insertar un nuevo registro en la tabla Alumnos con los valores proporcionados.

. Pasos a reproducir:

Probar este código en SQL

```
INSERT INTO Alumnos (id_alumno, nombre, apellido, n°legajo, contraseña, id_cursos)
VALUES (0001,'abel','villegas',001,'abel2005',01);
```

. Resultado esperado: Se debe insertar un nuevo registro en la tabla Alumnos con los valores proporcionados.

Prioridad: Alta

. CasoDePrueba2

. Descripción: Insertar un nuevo curso en la tabla Cursos.

. Precondiciones: No debe existir ningún registro en la tabla Cursos con el id_cursos igual a 01.

Postcondiciones: Se debe insertar un nuevo registro en la tabla Cursos con los valores proporcionados.

. Pasos a reproducir:

Probar este código en SQL

```
INSERT INTO Cursos (id_cursos,id_materias)
VALUES (01,02);
```

. Resultado esperado: Se debe insertar un nuevo registro en la tabla Cursos con los valores proporcionados.

. Prioridad: Media

. CasoDePrueba3

. Descripción: Insertar un nuevo libro en la tabla Libro.

. Precondiciones: No debe existir ningún registro en la tabla Libro con el id_libro igual a 1.

Postcondiciones: Se debe insertar un nuevo registro en la tabla Libro con los valores proporcionados.

. Pasos a reproducir:

Probar este código en SQL

```
INSERT INTO Libro (id_libro, nombre, autor, manual, literario, materias, id_materias, id_profesor)
```

```
VALUES (1, 'Curso de Matemáticas', 'John Smith', 'Sí', 'No', 'Matemáticas', 02, 013);
```

. Resultado esperado: Se debe insertar un nuevo registro en la tabla Libro con los valores proporcionados.

. Prioridad: Alta

.CasoDePrueba4

. Descripción: Insertar una nueva materia en la tabla Materias.

. Precondiciones: No debe existir ningún registro en la tabla Materias con el id_materias igual a 02.

Postcondiciones: Se debe insertar un nuevo registro en la tabla Materias con los valores proporcionados.

. Pasos a reproducir:

Probar este código en SQL

```
INSERT INTO Materias (id_materias, nombre, cursos)
```

```
VALUES (02, 'Curso de Matemáticas', 'septimo');
```

. Resultado esperado: Se debe insertar un nuevo registro en la tabla Materias con los valores proporcionados.

. Prioridad: Baja

. CasoDePrueba5

. Descripción: Insertar un nuevo curso nombrado en la tabla Nom_cursos.

. Precondiciones: No debe existir ningún registro en la tabla Nom_cursos con el nom_cursos igual a 'sexto'.

Postcondiciones: Se debe insertar un nuevo registro en la tabla Nom_cursos con los valores proporcionados.

. Pasos a reproducir:

Probar este código en SQL

```
INSERT INTO Nom_cursos (nom_cursos, nombre, id_profesores, id_cursos)
```

```
VALUES ('sexto', 'Curso de Matemáticas', 018, 01)
```

. Resultado esperado: Se debe insertar un nuevo registro en la tabla Nom_cursos con los valores proporcionados.

. Prioridad: Media

. CasoDePrueba6

. Descripción: Insertar un nuevo profesor en la tabla Profesores.

. Precondiciones: No debe existir ningún registro en la tabla Profesores con el id_profesores igual a 018.

Postcondiciones: Se debe insertar un nuevo registro en la tabla Profesores con los valores proporcionados.

.Pasos a reproducir:

Probar este código en SQL

```
INSERT INTO Profesores (id_profesores, nombre, apellidos, dni, materias, contraseña, id_materias)
```

```
VALUES (018, 'Juan', 'Gomez', 44740499, 'Matematicas', 'juan2003', 02);
```

. Resultado esperado: Se debe insertar un nuevo registro en la tabla Profesores con los valores proporcionados.

. Prioridad: Alta