## Días.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | List<Dia> obtenerDias(int idGrupo) | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| idGrupo: 18. | Ninguna excepción relativa a la persistencia, | Ninguna excepción. | El id del grupo corresponde a un grupo que existe o no. | Lista válida. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | boolean agregarDia(Dia dia) | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| Día:   * Día: Lunes. * HoraFIn: 20:00. * HoraInicio: 19:00. * Id: 0. * idTipo: 5. * Salón: Y. * Tipo: true. | True. | True. | El día tienes establecidos todos sus atributos correctamente. | True. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | boolean editarDia(Dia dia) | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| Día:   * Día: Viernes. * HoraFIn: 09:30. * HoraInicio: 09:00. * Id: 35. * idTipo: 18. * Salón: X. * Tipo: true. | True. | True. | El día existe en base de datos y tiene sus atributos establecidos correctamente. | True. |
| Día:   * Día: Viernes. * HoraFIn: 09:30. * HoraInicio: 09:00. * Id: 0. * idTipo: 18. * Salón: X. * Tipo: true. | False. | False. | El día con el id 0 no existe. | False. |

## Grupos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | List<Grupo> obtenerGruposProfesor(int idProfesor) | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| * idProfesor: 1. | Ninguna excepción relativa a la persistencia. | Ninguna excepción. | * El profesor con id 1 existe en base de datos. | Lista válida. |
| * idProfesor: 0. | Null Pointer Exception. | Null Pointer Exception. | * El profesor con id 0 no existe. | Null Pointer Exception. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | List<Grupo> obtenerGrupos() | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| Ninguna. | Ninguna excepción relativa a la persistencia. | Ninguna excepción. | Existen o no grupos registrados. | Lista válida. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | boolean registrarGrupo(Grupo grupo) | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| Grupo:   * Nombre: Soy un grupo. * Danza: Un tipo de danza. * Id: 0. * Profesor: id: 1. * Dia:   + Dia: Miércoles.   + Id: 0.   + Inicio: 09:30.   + Fin: 11:00.   + Salón: X.   + Tipo: true.   + idTipo: 0. | True. | True. | * El grupo tiene sus atributos asignados correctamente * El día marcado tiene sus atributos correctos y no choca con otro horario en la base de datos. * El profesor con id 1 existe en base de datos. | True. |
| Grupo:   * Nombre: Soy un grupo. * Danza: Un tipo de danza. * Id: 0. * Profesor: id: 0. * Dia:   + Dia: Miércoles.   + Id: 0.   + Inicio: 09:30.   + Fin: 11:00.   + Salón: X.   + Tipo: true.   + idTipo: 0. | False. | False. | * El grupo tiene sus atributos asignados correctamente * El día marcado tiene sus atributos correctos y no choca con otro horario en la base de datos. * El profesor con id 0 no existe en base de datos. | False |
| Grupo:   * Nombre: Soy un grupo. * Danza: Un tipo de danza. * Id: 0. * Profesor: id: 0. * Dia:   + Dia: Lunes.   + Id: 0.   + Inicio: 04:00.   + Fin: 15:00.   + Salón: X.   + Tipo: true.   + idTipo: 0. | Horario Exception. | Horario Exception. | * El grupo tiene sus atributos asignados correctamente * El día marcado choca con otro horario en la base de datos. * El profesor con id 1 existe en base de datos. | Horario Exception. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | List<Grupo> obtenerGruposAlumno(int idAlumno) | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| N/A. | Ninguna excepción relativa a la persistencia. | Ninguna excepción. | * Existe al menos un profesor registrado. | Lista válida. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Función** | boolean registrarAsistencia(Asistencia asistencia) | | | |
| **Entradas** | **Salidas esperadas** | **Salidas obtenidas** | **Condiciones de salida** | **Resultado** |
| Asistencia:   * Fecha: actual. * idAsistencia: 0. * idGrupo: primer id de los registros. * Alumnos: lista de alumnos del grupo. | True. | True. | * Existe al menos un grupo registrado. * El grupo tiene al menos un alumno inscrito. * Existe al menos un registro de asistencia relacionado con la fecha y el grupo. | True. |
| Asistencia:   * Fecha: actual. * idAsistencia: 0. * idGrupo: 0. * Alumnos: lista de alumnos del grupo. | False. | False. | * El grupo con id 0 no existe. | False. |
| Asistencia:   * Fecha: actual. * idAsistencia: 0. * idGrupo: primer id de los registros. * Alumnos: lista de alumnos vacía. | False. | False. | * La lista de alumnos está vacía. | False. |