AlumnoServicio.crearHorarioAlumno(Alumno alumno, HorarioServicio horario)
Siempre va a devolver un ArrayList <ArrayList>> de 5*6, en la tabla lo llamaremos horario

TestNum	Entradas	Salida esperada	Comentario
1	Alumno sin asignaturas	horario relleno de false	
2	Alumno con alguna asignatura	horario relleno de false excepto en las horas de las asignaturas que tiene que almacenará dicha asignatura	
3	Alumno con asignatura que tiene dos horarios coincidentes	horario relleno de false excepto en las horas de la asignatura que tiene en las que almacenará dicha asignatura, no debería afectar que tenga un horario repetido	Si una asignatura tiene dos horarios el lunes a primera por ejemplo, esa casilla se marcará con esa asignatura, no tendrá ningún efecto
4	Alumno con asignatura sin horarios	horario relleno de false	
5	Alumno con asignatura fuera de horario, es decir, día superior a 5 o inferior a 1, hora superior a 6 o inferior a 1	horario relleno de false	

ProfesorServicio.obtenerConvalidacionesPendientes(AlumnoServicio alumnoServicio, Long id)

Siempre va a devolver un ArrayList<SituacionExcepcional>, que en la tabla llamaremos lista

TestNum	Entradas	Salida esperada	Comentario
1	Id de Alumno con SituacionExcepcional	lista que contiene la SituacionExcepcional del Alumno	Replica el funcionamiento más deseado en la aplicación junto con el case 2
2	Id de Alumno sin SituacionExcecional	lista vacía	Replica el funcionamiento más deseado en la aplicación junto con el case 1
3	Un id de un alumno que no existe	lista vacía	El método ha tenido que ser corregido ya que fallaba cuando se le pasaba un id que no pertenecía ningún Alumno

Al método se le añadió una condición para que solo ejecute su lógica si existe un Alumno con dicho id, si no existe devolverá una lista vacía.

UsuarioServicio.autogenerarCodigo()

TestNum	Entradas	Salida esperada	Comentario
1	usuario. getCodigoInvitacion()	un tipo de dato int	Comprueba que el tipo de dato generado es un int
2	usuario. getCodigoInvitacion()	un tipo de dato String	Comprueba que el tipo de dato generado no es un String
3	usuario. getCodigoInvitacion()	se espera que de true al estar dentro del rango	Comprobar código entre 1 y 1 millón, se prueba el número 500
4	usuario. getCodigoInvitacion()	se espera que de false al estar fuera del rango	Comprobar código entre 1 y 1 millón, se prueba el número 1000005
5	usuario. getCodigoInvitacion()	se espera que de false al estar fuera del rango	Comprobar código entre 1 y 1 millón, se prueba el número 500

UsuarioServicio.aceptarValidacion(int codigo, String contrasenya)

TestNum	Entradas	Salida esperada	Comentario
1	código	si ambos coinciden, debe devolver true	Comprobar que el código pasado coincide con el generado
2	código	si no coinciden, debe devolver false	Comprobar que si el código pasado es incorrecto, no funcione
3	lista vacía	al estar la lista vacía, devuelve false	Comprobar si falla al pasarle una lista vacía de usuarios
4	codigo, contraseña	debería devolver true si se ha cambiado correctamente la contraseña	Comprobar que el método cambia la contraseña por la nueva.

ProfesorServicio.negarConvalidacion(*Long id, SituacionExcepcionalServicio situacionExcepcionalServicio,AlumnoServicio alumnoServicio, AsignaturaServicio asignaturaServicio*) → Comprobamos que el método responde a una llamada, ya que su tipo de retorno es Void, y que actualiza los campos de las instancias con las que se relaciona la instancia de SituacionExcepcional (Alumno y Asignatura):

TestNum	Entradas	Salida esperada	Comentario
1	Instancias de: Un número de tipo Long (1L), SituacionExcepcionalServicio, AlumnoServicio, AsignaturaServicio	Asertos: True	Los asertos que comprueban los parámetros que se pasan al método deben pasarse, devolviendo True cada uno.
2	Las instancias necesarias para ejecutar el método.	Asertos: True	Comprueba que el método actualiza el campo “situacionesExc” de la entidad Alumno y Asignatura.

-**UsuarioRepositorio.findByEmail**(String email); → Pablo

TestNum	Entradas	Salida esperada	Comentario
1	correo (string)	usuario	Devolverá el usuario completo según el correo

-Algún **UsuarioRepositorio.findAll()** de cualquiera. → Pablo

TestNum	Entradas	Salida esperada	Comentario
1	nada	listado usuario	Devuelve todo el listado de usuarios que haya

-Algún **UsuarioRepositorio**.**findById(id)** de cualquiera.→ Pablo

TestNum	Entradas	Salida esperada	Comentario
1	int (id)	usuario	Devuelve el usuario cuyo id sea igual al introducido