

### Diseño de casos de pruebas unitarias

Clase: Agenda.		Método: addStudent()	
Caso #	Escenario de prueba	Entradas	Salida
1	Agregar un nuevo estudiante	Datos del estudiante	Estudiante en la lista de contactos.
2	Agregar un estudiante que ya existe en la agenda	Datos repetidos	EL estudiante con dichos datos ya existe.

Clase: Agenda.		Método: removeStudent()	
Caso #	Escenario de prueba	Entradas	Salida
1	Eliminar un estudiante	Datos del estudiante	Estudiante eliminado de la lista de contactos.

Clase: Agenda.		Método: search()	
Caso #	Escenario de prueba.	Entradas	Salida
1	Buscar estudiante por su nombre	Nombre del estudiante	Datos del estudiante con el nombre buscado.
2	Buscar estudiante por su apellido.	Apellido del estudiante.	Datos del estudiante con el apellido buscado.
3	Buscar estudiante por el correo electrónico.	Correo electrónico del estudiante.	Datos del estudiante con el correo electrónico buscado.
4	Buscar estudiante por número de teléfono.	Número de teléfono del estudiante.	Datos del estudiante con el número de teléfono buscado.

Clase: Agenda.		Método: assignedSubjectsAverage()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Calcular el promedio de las materias matriculadas por los estudiantes.	Lista con las materias matriculadas por los estudiantes.	Valor decimal que representa el promedio de las materias.

Clase: Agenda.		Método: assignedCreditsAverage()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Calcular el promedio de los créditos de las materias de los estudiantes.	Créditos de todas las materias matriculadas.	Valor del promedio de los créditos de las materias.

Clase: Agenda		Método: mostAssignedSubject()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Calcular la materia más veces matriculada.	Lista con las materias matriculadas por todos los estudiantes.	Materia más matriculada por los estudiantes.
2	Hay más de una materia más veces matriculada.	En la lista hay dos materias más veces matriculadas.	Todas las materias más veces matriculadas.

Clase: Agenda		Método: lessAssignedSubject()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Calcular la materia menos veces matriculada.	Lista con todas las materias matriculadas por los estudiantes.	Materia menos matriculada por los estudiantes.
2	Existen dos materias menos matriculadas.	En la lista de materias hay dos materias con la menor cantidad de veces matriculadas.	Todas las materias menos veces matriculadas.

Clase: Agenda.		Método: assignedSubject()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Mostrar las materias matriculadas por los estudiantes.	Ninguna.	Todas las materias matriculadas por los estudiantes.
2	No hay materias por mostrar.	Ninguna.	No existen materias en la agenda.

Clase: Agenda.		Método: loadSubjects()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Permitir cargar un archivo con todas las materias.	Archivo de texto.	Todas las materias cargadas en la agenda.
2	Archivo vacío o dañado.	Archivo de texto	No es posible agregar las materias a la agenda

Clase: Agenda.		Método: loadSubjects()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Cargar una o varias materias nuevas.	Archivo con nuevas materias.	Nuevas materias en la agenda.
2	No se carga ninguna materia.	Archivo con nuevas materias.	No es posible agregar materias.

Clase: Agenda.		Método: saveStudents()	
Caso #	Escenario de prueba.	Entradas.	Salida.
1	Permite guardar los estudiantes en la agenda.	Estudiantes para guardar	Estudiantes guardados en la agenda.