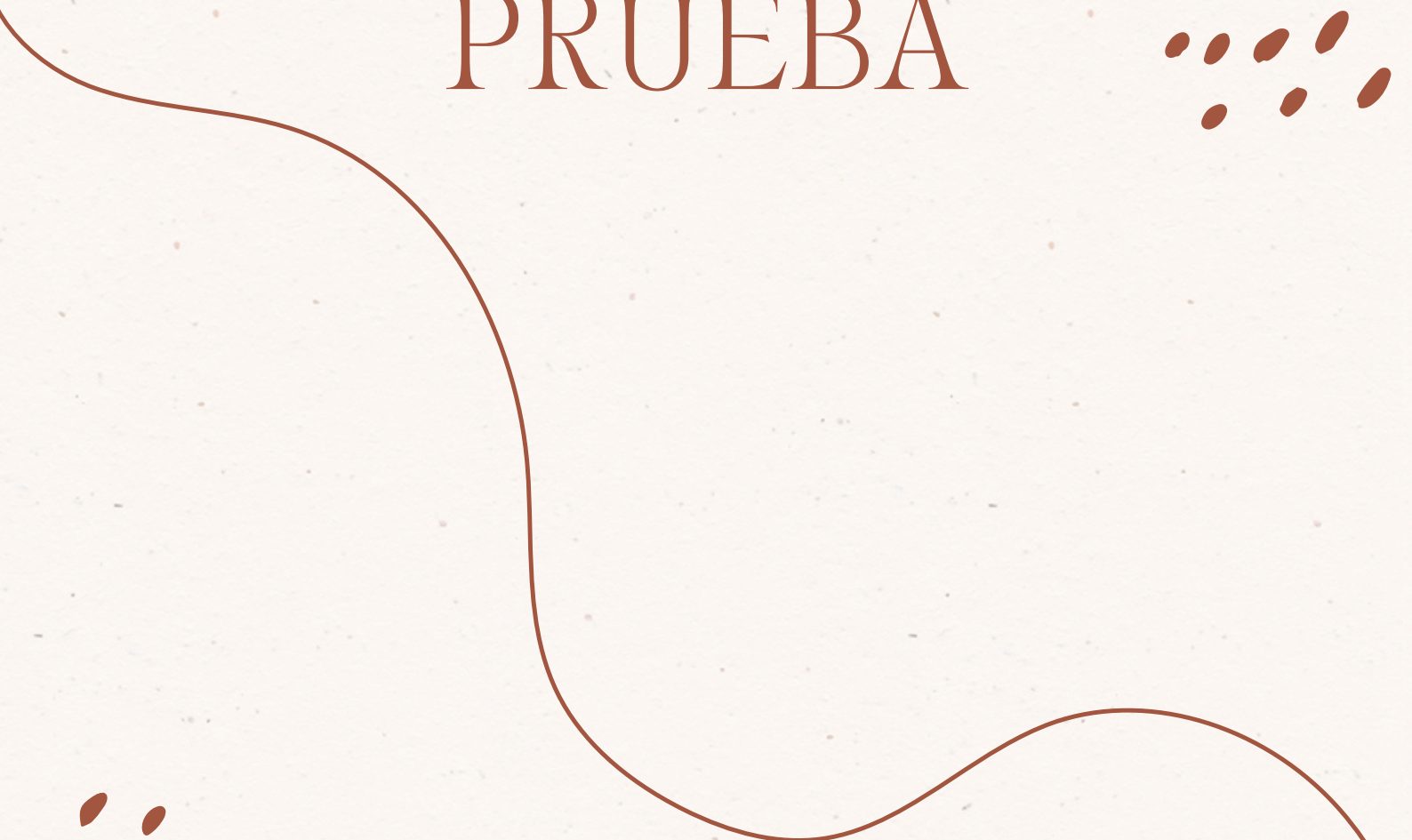
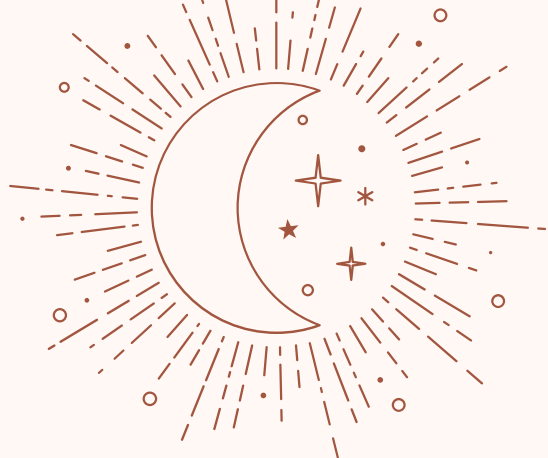




CASOS DE PRUEBA




MIRIAM RODRÍGUEZ ANTEQUERA



CLASE AUTOR

Nº	DESCRIPCIÓN	ENTRADA	RESULTADO ESPERADO
1	Constructor con nombre válido	"Autor Prueba 1"	Se crea correctamente un objeto Autor sin errores
2	Constructor con nombre vacío	""	Se lanza una excepción indicando que el nombre no puede estar vacío
3	setNomAutor con nombre válido	"Autor Modificado"	Se actualiza el nombre correctamente y devuelve true
4	setNomAutor con nombre vacío	""	No se actualiza el nombre y devuelve false
5	guardarAutor nuevo autor	"AutorNuevo Test"	El autor se guarda exitosamente en el sistema
6	listarAutores	-	Se obtiene una lista de autores con al menos un elemento
7	verAutor con ID válido	ID último autor creado	Se devuelve el objeto Autor con el ID correcto
8	eliminarAutor válido	ID último autor creado	El autor se elimina correctamente y devuelve true
9	eliminarAutor con ID inexistente	999999	No se encuentra el autor y devuelve false



Muestra de salida exitosa de cada una de las pruebas anteriormente explicadas

```
root@57248c2b7c83:/var/www/html# php ./tests/autorTest.php
==== INICIO DE PRUEBAS DE AUTOR ====
Constructor con nombre válido OK
Constructor con nombre inválido lanzó excepción
setNomAutor con nombre válido OK
setNomAutor inválido fue rechazado
guardarAutor nuevo autor OK
listarAutores OK (total: 13)
verAutor OK
Autor ID 29 eliminado correctamente.
eliminarAutor OK
Autor ID 28 eliminado correctamente.
AutorNuevoTest eliminado correctamente.
eliminarAutor inexistente fue correctamente rechazado
==== FIN DE PRUEBAS ====
```



CLASE USUARIO

Nº	DESCRIPCIÓN	ENTRADA	RESULTADO ESPERADO
1	Constructor válido	"UsuarioPrueba", "usuario@correo.com"	Se crea correctamente un objeto Usuario sin errores
2	guardarUsuario	UsuarioPrueba	El usuario se guarda exitosamente en el sistema
3	verUsuarioPorNom	"UsuarioPrueba"	Se devuelve el objeto Usuario con el nombre correcto
4	verUsuarioPorEmail	"usuario@correo.com"	Se devuelve el objeto Usuario con el email correcto
5	actualizarDatos	Email y foto nuevos	Los datos del usuario se actualizan correctamente y devuelve true
6	cambiarClave	Nueva clave	La contraseña se modifica correctamente y devuelve true
7	verificarLogin	Nuevas credenciales	Se verifican las credenciales y se devuelve el objeto Usuario válido
8	eliminarUsuario	"UsuarioPrueba"	El usuario se elimina correctamente y devuelve true

Muestra de salida exitosa de cada una de las pruebas anteriormente explicadas

```
root@72412dd6e18b:/var/www/html# php ./tests/usuarioTest.php
==== INICIO DE PRUEBAS DE USUARIO ====
Constructor con datos válidos OK
guardarUsuario nuevo usuario OK
verUsuarioPorNom OK
verUsuarioPorEmail OK
actualizarDatos OK
cambiarClave OK
verificarLogin OK
Usuario eliminado correctamente.
eliminarUsuario OK
==== FIN DE PRUEBAS ====
```



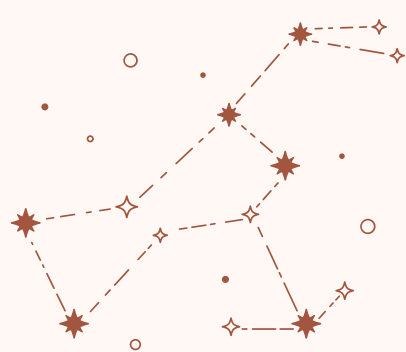


CLASE GÉNERO

Nº	DESCRIPCIÓN	ENTRADA	RESULTADO ESPERADO
1	Constructor válido	"Género Prueba 1"	Se crea correctamente un objeto Género sin errores
2	Constructor inválido	""	Se lanza una excepción indicando que el nombre no puede estar vacío
3	listarGeneros	-	Se obtiene una lista de géneros con al menos un elemento
4	verGenero con ID válido	ID de un género	Se devuelve el objeto Género con el ID correcto

Muestra de salida exitosa de cada una de las pruebas anteriormente explicadas

```
root@72412dd6e18b:/var/www/html# php ./tests/generoTest.php
==== INICIO DE PRUEBAS DE GÉNERO ====
Constructor con nombre válido OK
Constructor con nombre inválido lanzó excepción
listarGeneros OK (total: 14)
verGenero OK
==== FIN DE PRUEBAS ====
```

CLASE GRUPO

N°	DESCRIPCIÓN	ENTRADA	RESULTADO ESPERADO
1	Constructor con usuario válido	"Género Prueba 1"	Se crea correctamente un objeto Grupo sin errores
2	guardarGrupo	Objeto Grupo	El grupo se guarda exitosamente en el sistema
3	agregarLibro	ID libro 1	El libro se agrega al grupo correctamente y devuelve true
4	getLibros	-	Se obtiene una lista de libros con al menos un elemento
5	eliminarLibro	ID libro 1	El libro se elimina correctamente y devuelve true
6	eliminarGrupo	ID grupo creado	El grupo se elimina correctamente y devuelve true

Muestra de salida exitosa de cada una de las pruebas anteriormente explicadas

```
root@72412dd6e18b:/var/www/html# php ./tests/grupoTest.php
==== INICIO DE PRUEBAS DE GRUPO ====
Usuario líder guardado OK
Constructor con datos válidos OK
guardarGrupo nuevo grupo OK
agregarLibro OK
getLibros OK (total: 1)
eliminarLibro OK
Grupo Prueba eliminado correctamente.
==== FIN DE PRUEBAS ====
```

CLASE LIBRO

Nº	DESCRIPCIÓN	ENTRADA	RESULTADO ESPERADO
1	Constructor válido	"Libro Prueba"	Se crea correctamente un objeto Libro sin errores
2	verLibro con ID válido	1	Se devuelve el objeto Libro con el ID correcto
3	listarLibros	-	Se obtiene una lista de libros con al menos un elemento
4	guardarLibro	"LibroNuevoTest"	El libro se guarda exitosamente en el sistema
5	getAutores y getGeneros	Libro ID 1	Se obtienen listas de autores y géneros con al menos un elemento cada una
6	eliminarLibro	ID del nuevo libro	El libro se elimina correctamente y devuelve true

Muestra de salida exitosa de cada una de las pruebas anteriormente explicadas

```
root@57248c2b7c83:/var/www/html# php ./tests/libroTest.php
==== INICIO DE PRUEBAS DE LIBRO ====
Constructor con datos válidos OK
verLibro OK
listarLibros OK (total: 18)
guardarLibro nuevo libro OK
cargarAutores OK (total: 1)
cargarGeneros OK (total: 1)
Libro ID 21 eliminado.
eliminarLibro OK
==== FIN DE PRUEBAS ====
```




FIN

