**Proyecto: BiblioMóvil**

**Plan de equivalencia**

**Paquete: Dominio**

* **Clase: BiblioMovil**

-Método descripciónLibro(String titulo)

**Particiones equivalencias:** [Libro encontrado],[Libro no encontrado]

**Valores límites:** No existen.

**Valores posibles:** Cien años de soledad, libro Inventado.

-Método registrarSocio(Socio nuevoSocio),getSocios();

Se hizo en conjunto para saber si se añadían los socios al array.

**Particiones equivalencias:** [Socio añadido],[Socio no añadido]

**Valores límites** No existen.

**Valores posibles:** Socio nuevoSocio=new Socio("Juan","González","14-07-1996","juang@gmail.com","123456"), no añadir socio.

-Método Comprobar\_email(String email)

**Particiones equivalencias:** [Email coincide],[Email no coincide]

**Valores límites:** No existen.

**Valores posibles:** [villaltamaria@gmail.com](mailto:villaltamaria@gmail.com), noexiste@gmail.com

- Método Comprobar\_contrasenya(String contrasenya)

**Particiones equivalencias:** [Contraseña coincide],[Contraseña no coincide]

**Valores límites:** No existen.

**Valores posibles:**123456, incorrecto

-Método tituloLibro(String nombre)

**Particiones equivalencias:** [Libro existe en biblioteca],[Libro no existe en biblioteca]

**Valores límites:** No existen.

**Valores posibles:** Cien años de soledad, Este Libro No Existe.

-Método getDisponibilidad(String titulo)

**Particiones equivalencias:** [Disponible],[No disponible]

**Valores límites:** No existen.

**Valores posibles:** Cien años de soledad, La chica de nieve.

-Método Libro Libro(String titulo)

**Particiones equivalencias:** [Objeto Libro Existe],[Objeto Libro No Existe]

**Valores límites:** No existen.

**Valores posibles:** Cien años de soledad, Matemática 4

* **Clase: Libro**

-Método getTitulo

**Particiones equivalencia:** [“Devuelve un título”]

**Valores limites:** No existen

**Valores posibles:** Lugares asombrosos.

-Método getAnyo\_publicacion

**Particiones equivalencia:** [“Devuelve el año de publicación”]

**Valores limites:** No existen

**Valores posibles**: 2019.

-Método getAutor

**Particiones equivalencia:** [“Devuelve el nombre del autor”]

**Valores limites:** No existen

**Valores posibles**: Luis Villar.

-Método getEditorial

**Particiones equivalencia:** [“Devuelve la editorial”]

**Valores limites:** No existen

**Valores posibles:** ALFAGUARA JUVENIL.

-Método getDisponibilidad

**Particiones equivalencia:** [Disponible] [No disponible]

**Valores limites:** No existen.

**Valores posibles:** Libro mi\_libro1 = new Libro("Lugares asombrosos", 2019, "Luis Villar", "ALFAGUARA JUVENIL", true, Formato.FISICO, "La curiosidad es el motor de las aventuras que lo lanzan hasta los terrenos más inhospitos y misteriosos.") **🡪Disponible** , Libro mi\_libro2 = new Libro("Don Quijote de la Mancha", 1615, "Miguel de Cervantes", "Francisco de Robles", false, Formato.DIGITAL, "Es la primera obra genuinamente desmitificadora de la tradición caballeresca y cortés por su tratamiento burlesco.") **🡪No disponible**

-Método getFormato

**Particiones equivalencia:** [Formato Físico], [Formato Digital]

**Valores limites:** No existen

**Valores posibles**: Libro mi\_libro1 = new Libro("Lugares asombrosos", 2019, "Luis Villar", "ALFAGUARA JUVENIL", true, Formato.FISICO, "La curiosidad es el motor de las aventuras que lo lanzan hasta los terrenos más inhospitos y misteriosos.") **🡪Formato Físico** , Libro mi\_libro2 = new Libro("Don Quijote de la Mancha", 1615, "Miguel de Cervantes", "Francisco de Robles", false, Formato.DIGITAL, "Es la primera obra genuinamente desmitificadora de la tradición caballeresca y cortés por su tratamiento burlesco.") **🡪Formato Digital**

-Método getDescripcion

**Particiones equivalencia:** [“Devuelve la descripción del libro”]

**Valores limites:** No existen

**Valores posibles**: Libro mi\_libro1 = new Libro("Lugares asombrosos", 2019, "Luis Villar", "ALFAGUARA JUVENIL", true, Formato.FISICO, "La curiosidad es el motor de las aventuras que lo lanzan hasta los terrenos más inhospitos y misteriosos.") **🡪Descripción** “La curiosidad es el motor de las aventuras que lo lanzan hasta los terrenos más inhospitos y misteriosos”

* **Clase: Prestamo**

-Método anyadirPrestamo(Prestamo prestamo), getLibrosPrestados():

Se hizo en conjunto para saber si se añadían los prestamos al array.

**Particiones equivalencias:** [Prestamo añadido],[Prestamo no añadido]

**Valores límites:** No existen.

**Valores posibles:** Prestamo nuevoPrestamo=new Prestamo(Prestamo.fecha\_inicio(2020, Calendar.JANUARY, 4),Prestamo.fecha\_fin(2020,Calendar.JANUARY,4),"chaconkevim@gmail.com","Mientras te olvido"), no añadir prestamo.

-Método devolverLibro(String emailSocio,String titulo)

Se hizo en conjunto para saber si se añadían los prestamos al array.

**Particiones equivalencias:** [Prestamo devuelto],[Prestamo no devuelto]

**Valores límites:** No existen.

**Valores posibles:** Prestamo nuevoPrestamo=new Prestamo(Prestamo.fecha\_inicio(2020, Calendar.JANUARY, 4),Prestamo.fecha\_fin(2020,Calendar.JANUARY,4),"chaconkevim@gmail.com","Mientras te olvido"), no devolver prestamo.

-Método devolverLibro(String emailSocio,String titulo), getLibrosPrestados():

Se hizo en conjunto para saber si se eliminaban los prestamos al array.

**Particiones equivalencias:** [Prestamo eliminado],[Prestamo no eliminado]

**Valores límites:** No existen.

**Valores posibles:** jonh612@outlook.com, no devolver prestamo.

-Método devolverLibro(String emailSocio,String titulo)

Se hizo en conjunto para saber si se añadían los prestamos al array.

**Particiones equivalencias:** [Prestamo devuelto],[Prestamo no devuelto]

**Valores límites:** No existen.

**Valores posibles:**ArrayList con prestamo devuelte(un elemento menos), no eliminar el prestamo.

-Método getEmailSocio,setEmailSocio(String emailSocio)

**Particiones equivalencias:** [Email establecido], [Email no deseado],

**Valores límites:** No existen.

**Valores posibles:**villaltamaria@gmail.com,null

-Método obtenerFechaPrestamo(String emailSocio,String titulo)

EMAIL

**Particiones equivalencias:** [Email existente], [Email no existente],

**Valores límites:** No existen.

**Valores posibles:**villaltamaria@gmail.com, estemailnoexiste@gmail.com.

TITULO

**Particiones equivalencias:** [Titulo existente], [Titulo no existente],

**Valores límites:** No existen.

**Valores posibles:**Cien años de soledad,La chica de nieve

-Método tituloCoincide(String email,String titulo)

EMAIL

**Particiones equivalencias:** [Email coincide],[Email no coincide]

**Valores límites:** No existen.

**Valores posibles:** villaltamaria@gmail.com, [estemailnoexiste@gmail.com](mailto:estemailnoexiste@gmail.com).

TITULO

**Particiones equivalencias:** [Titulo coincide], [Titulo no coincide]

**Valores límites:** No existen.

**Valores posibles:** Cien años de soledad, Este libro no está en préstamo.

-Método obtenerFechaPrestamo(String emailSocio,String titulo)

EMAIL

**Particiones equivalencias:** [Email coincide],[Email no coincide]

**Valores límites:** No existen.

**Valores posibles:** villaltamaria@gmail.com, [estemailnoexiste@gmail.com](mailto:estemailnoexiste@gmail.com).

TITULO

**Particiones equivalencias:** [Titulo coincide], [Titulo no coincide]

**Valores límites:** No existen.

**Valores posibles:** Cien años de soledad, Este libro no está en préstamo.

* **Clase: Sanción**

-Método miSancion()

**Particiones equivalencias:** [Objeto Sancion existe], [Objeto Sancion no existe]

**Valores límites:** No existen.

**Valores posibles:** Objeto exista en sanción como: Sancion=new Sancion("zapatacamila@gmail.com","Cien años de soledad"), Objeto no exista en sanción como: Sancion=new Sancion("villaltamaria @gmail.com","Cien años de soledad")

-Método comprobarSancion(String email)

**Particiones equivalencias:** [Existe Sancion], [No Existe Sancion]

**Valores límites:** No existen.

**Valores posibles:** zapatacamila@gmail.com, villaltamaria@gmail.com

-Método anyadirSancion(Sancion sanción), getMis\_sanciones()

Se hizo en conjunto para saber si se añadían los prestamos al array.

**Particiones equivalencias:** [Sancion añadido],[Sancion no añadido]

**Valores límites:** No existen.

**Valores posibles:** Sancion nuevaSancion=new Sancion("lau\_aguirre@hotmail.com","La chica de nieve"), no añadir prestamo.

* **Clase: Usuario**

**Particiones equivalencias:** [Usuario registrado], [Introducir datos erróneos]

nota: Este método siempre va a devolver un socio, porque está tratado para que el usuario si o si se registre, cuando pone un dato erróneo se lo volverá a pedir con el fin de que se pueda realizar el registro exitosamente.

**Valores límites:** No existen.

**Valores posibles:** BiblioMovil miBiblio.

* **Clase: Socio**

En todos los métodos se usa el objeto *Socio mi\_socio= new Socio("Antonella", "Gutierrez","12-12-1999", "anto123@gmail.com", "anto123"))*, y de allí hacemos las pruebas.

-Método getNombre

**Particiones equivalencia:** [“Devuelve el nombre del socio”].

**Valores limites:** No existen

**Valores posibles:** Antonella.

-Método getApellidos

**Particiones equivalencia:** [“Devuelve el apellido del socio”].

**Valores limites:** No existen

**Valores posibles:** Gutierrez.

-Método getFecha\_nacimiento

**Particiones equivalencia:** [“Devuelve la fecha de nacimiento del socio”].

**Valores limites:** No existen

**Valores posibles:**: 12-12-1999.

-Método getEmail

**Particiones equivalencia:** [“Devuelve el Email del socio”]

**Valores limites:** No existen

**Valores posibles:** anto123@gmail.com.

-Método getContrasenya

**Particiones equivalencia:** [“Devuelve la contraseña del socio”]

**Valores limites:** No existen

**Valores posibles:** anto123