Requerimiento Funcional	Nombre de la Clase	Nombre del método
R1- register user	Controller	+registerUser(String id, String name, int userType):boolean
	Executable	+registerBibliographicProduct():void
	Premium	Premium (String id, String name)
	Regular	Regular (String id, String name)
R2- register bibliography products	Controller	+registerBibliographicProduct(String id,String name,int pagesNumber, int genre, int day, int month, int year, int productType, String review, String url, double value, double suscriptionValue, String periodicEmision): boolean +validateIDHexadecimal(String id):boolean +validateIDnumeric(String id):boolean
	Executable	+registerBibliographicProduct():void
	Book	+ Book(String id, String name, int pagesNumber, String review, Calendar publicDate, Genre genre, String url, double value)
	Magazine	Magazine(String id, String name, int pagesNumber, String periodicEmision, Calendar publicDate, GenreMagazine genre, String url, double suscriptionValue)
R3- Modify bibliographic product	Controller	editBibligrapicProduct(int userPosition, String newInfo, int change):boolean getBibliographicPList(): String

	BibliographicProduct	.setName(string newInfo)
		setUrl(string newInfo)

	Magazine	setPeriodicEmision(string newInfo)
		setSuscriptionValue(double newInfo)

	Book	setReview(string newInfo)
		setValue(double newInfo)

	Executable	editBibligrapicProduct():void
R4- Delete bibliographic product	Executable	deleteBibliographicProduct():void
	Controller	getBibliographicPList():String deleteBibliographicProduct(int userPosition): boolean
R5- buy bibliographic product	Controller	buyBibliographicProduct(int option, int day,int month,int year, int userOP):boolean generateBill(int option, int day,int month,int year): String getBibliographicPList(): string
	Executable	buyBibliographicProduct():void
	Regular	getRegularBooks(): RegularBooks[] getRegularMagazine(): RegularMagazine[]

	Book	getReview();
		getValue();
		getGenre();

	Premium	getPremiumLibrary()
	Magazine	getPeriodicEmision();

		getSuscriptionValue();
		getGenreM();

	BibliographicProduct	getId(),
		getName()
		getPagesNumber()

R6-Virtual library
.
	Controller	showUserLibrary(int userPosition)
		showPremiumLibrary(int userPosition)
		runMatrixPremium(String action, int userP)
		runRegularMatrix(String action, int userP)
		showRegularLibrary(int userPosition)
		libraryTRY(int userPosition)

	Executable	consultLibrary()

	Premium	sortLibraryByPublishingDate()
		showPremiumLibrary()
		addProductsMatrix()
		public boolean selectProductByIdentifier(String identifier)
		selectProductByCoordinates(int x, int y)
		nextPage()
		previousPage()
		createLibraryPages()

	Regular	setFullLibrary()
		selectProductByIdentifier(String identifier)
		selectProductByCoordinates(int x, int y)
		showMatrixLibrary()
		addProductsMatrix()
		sortBooksByPublicationDate();

R7- Reading session	Executable	readingDEMO()

	Controller	showUserLibrary(int userPosition)
		showPremiumLibrary(int userPosition)
		showRegularLibrary(int userPosition)
		createBookSim(int userOP, int productOP)
		trySimulation(int userOP, int productOP, int action)
		mthd2(int action, ReadingSimulation RP)
		evaluateAnnounce(int pagesRead)
		readingSimRegular(int userOP, int productOP, int action)
		getTypeUser(int userOP)
		getProductSimulationP(String id, int userOP)

		getProductSimulationR(String id, int userOP)

	Premium	getPremiumLibrary
		addReadingSesion(String productID, String productName)
		showPremiumLibrary()

	Regular	setFullLibrary()
		addReadingSesion(String productID, String productName)
		showMatrixLibrary()

R8- generate reports	Executable	reporNumberPagesReadByBibliographicP()
		soldProductsByGender()

	Controller	reporNumberPagesReadByBibliographicP()
		soldProductsByGender()

	Regular	setFullLibrary() regularSimulation
	Premium	getPremiumLibrary() readingSimulation