

Requerimiento Funcional	Nombre de la Clase	Nombre del método
● R1: Ingresar una comunidad	Class Executable	<ul style="list-style-type: none"> <li>Executable()</li> <li>administratorMenu()</li> </ul>
	Class Controller	<ul style="list-style-type: none"> <li>addCommunity(nameCommunity : String , typeCommunity : string, nameRepresentative : String, representativeContact..)</li> <li>findCommunity()</li> </ul>
	Class Community	community(): Community
	Class Product	product(): Product
● R2: Ingresar un lugar	Class Executable	<ul style="list-style-type: none"> <li>administratorMenu()</li> </ul>
	Class Controller	<ul style="list-style-type: none"> <li>addPlace(placeName : String, departamenThereIsLocated : String, extensionKm : double, typePlace : String, inagurationPlace : String, placelImage : String, communityName : String, economicResources : double, speciesInhabits : String) : String</li> <li>findPlace(): String</li> </ul>
	Class Place	- place(): Place
● R3: Ingresar producto a la comunidad	Class Executable	- enterSpecie() : void
	Class Controller	<ul style="list-style-type: none"> <li>enterSpecies(specieName : String, specieType : String, specielImage : String, numerCopies : int, species association : String) : String</li> <li>findPlace</li> </ul>
	Class Product	- product(): Product
● R4: Eliminar producto de la comunidad	Class Executable	- deleteProduct() : void
	Class Controller	- deleteProductFromCommunity(identif yProduct : String, DeleteProductName : String)
	Class Community	<ul style="list-style-type: none"> <li>Community() : Community</li> <li>deleteProduct(products : Product) : void</li> </ul>

● <b>R5: Ingresar una especie a un lugar</b>	Class Executable	- enterSpecie() : void
	Class Controller	- enterSpecies(specieName : String, specieType : String, specieImage : String, numerCopies : int, species association : String) : String
● <b>R6: Modificar especie a un lugar</b>	Class Executable	- modifyDataSpecies() : void
	Class Controller	- modifySpecies(namePlace : String, modifySpeciesName : String, newData)
	Class Specie	- specie(): Specie - updateSpecieInfo(name : String, typeSpecie : String, photoAnimal : String, numberSpecimens : int, associationWithPlaces : String)
● <b>R7: Consultar la información de un lugar</b>	Class Executable	- consultPlaceInfo() : void
	Class Controller	- consultLocalInfo(identifyPlaceName : String, identifyDepartament : String)
	Class Place	- getName() : String
● <b>R8: Consultar la información de las comunidades de un departamento</b>	Class Executable	- +consultCommunityDepartmentInfo() : void
	Class Controller	- consultLocalInfo(identifyPlaceName : String, identifyDepartament : String) : void
	Class Place	- getName() : String
● <b>R9: Consultar la información de las comunidades de un departamento</b>	Class Executable	- consultPlaceInfo() : void
	Class Controller	- consultLocalInfo(identifyPlaceName : String, identifyDepartament : String)
	Class Community	- community Community
● <b>R10: Consultar la información de las comunidades con problemas específicos</b>	Class Community	- getMainProblems() : ArrayList < String >
● <b>R11: Consultar el nombre del lugar con</b>	Class Executable	- consultPlaceLargestNumber() : void

<b>mayor cantidad de especies</b>		
	Class Controller	- showPlaceWithMostSpecies() : String
	Class Place	- calculateLargestNumberSpecies() : String
<b>• R12: Consultar los tres lugares más grandes por kilómetro cuadrado</b>	Class Executable	- consultThreeLargestKmPlaces() : void
	Class Controller	- showThreeLargestPlaces() : String
	Class Place	- getExtensionKm() : double