Diseño de pruebas de los servicios

Pruebas sobre la implementación de com.taller1.service.implementation.ProductServiceImpl			
PRUEBAS UNITARIA	PRUEBAS UNITARIAS		
Método a probar	Paramétro(s)	Resultado esperado (Prueba unitaria)	
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	null, null, null	El método lanza NullPointerException con mensaje "Producto no puede ser nulo". No hay llamado a ningún método de prodRepository (mock), pero hay llamados a los métodos findByld(Integer id) de prodCatRepository (mock) y prodSubRepository (mock).	
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	Instancia de Product, null, ID (Integer) de Instancia de Productsubcategory, Instancia de Unitmeasure	El método lanza NullPointerException con mensaje "No escribió un id de la categoria existente para producto". No hay llamado a ningún método de prodRepository (mock), pero hay llamados a los métodos findByld(Integer id) de prodCatRepository (mock) y prodSubRepository (mock).	
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	Instancia de Product, ID (Integer) de Instancia de Productcategory, null, Instancia de Unitmeasure	El método lanza NullPointerException con mensaje "No escribió un id de la subcategoria existente para producto". No hay llamado a ningún método de prodRepository (mock), pero hay llamados a los métodos findByld(Integer id) de prodCatRepository (mock) y prodSubRepository (mock).	
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	Instancia de Product, ID (Integer) de Instancia de Productcategory, ID (Integer) de Instancia de Productsubcategory, null	El método lanza NullPointerException con mensaje "No escribió el id de unidad de medida del producto". No hay llamado a ningún método de prodRepository (mock), pero hay llamados a los métodos findByld(Integer id) de prodCatRepository (mock) y prodSubRepository (mock).	
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	Instancia de Product con Productnumber="0" y el resto de los atributos respetando las condiciones dadas en el enunciado, ID (Integer) de Instancia de Productcategory, ID (Integer) de Instancia de Productsubcategory, Instancia de Unitmeasure	El método lanza Exception con mensaje "Product number must be greater than 0". No hay llamado a ningún método de prodRepository (mock), pero hay llamados a los métodos findByld(Integer id) de prodCatRepository (mock) y prodSubRepository (mock).	

saveProduct	Instancia de Product con	El método lanza NumberFormatException
(Product p,	Productnumber="A" y el resto de	con mensaje "For input string: "A"". No
Integer pCatId,	los atributos respetando las	hay llamado a ningún método de
Integer pSubId,	condiciones dadas en el enunciado,	prodRepository (mock), pero hay llamados
Unitmeasure um)	ID (Integer) de Instancia de	a los métodos findById(Integer id) de
,	Productcategory, ID (Integer) de	prodCatRepository (mock) y
	Instancia de Productsubcategory,	prodSubRepository (mock).
	Instancia de Unitmeasure	, , , , , ,
saveProduct	Instancia de Product con	El método lanza Exception con mensaje
(Product p,	Daystomanufacture=0 y el resto de	"Days to manufacture must be greater
Integer pCatId,	los atributos respetando las	than 0". No hay llamado a ningún método
Integer pSubId,	condiciones dadas en el enunciado,	de prodRepository (mock), pero hay
Unitmeasure um)	ID (Integer) de Instancia de	llamados a los métodos findById(Integer
Official casare arri	Productcategory, ID (Integer) de	id) de prodCatRepository (mock) y
	Instancia de Productsubcategory,	prodSubRepository (mock).
		prousubkepository (mock).
covo Dro de est	Instancia de Unitmeasure	El mátado lango Evocation con magazia
saveProduct	Instancia de Product con	El método lanza Exception con mensaje
(Product p,	Sellstartdate después de	"Sell end date is before sell start date". No
Integer pCatId,	Sellenddate y el resto de los	hay llamado a ningún método de
Integer pSubId,	atributos respetando las	prodRepository (mock), pero hay llamados
Unitmeasure um)	condiciones dadas en el enunciado,	a los métodos findById(Integer id) de
	ID (Integer) de Instancia de	prodCatRepository (mock) y
	Productcategory, ID (Integer) de	prodSubRepository (mock).
	Instancia de Productsubcategory,	
	Instancia de Unitmeasure	
saveProduct	Instancia de Product con	El método lanza NullPointerException con
(Product p,	Sellstartdate y Sellenddate nulos y	mensaje "Cannot invoke
Integer pCatId,	el resto de los atributos respetando	"java.sql.Timestamp.before(java.sql.Timest
Integer pSubId,	las condiciones dadas en el	amp)" because "Sellenddate" is null". No
Unitmeasure um)	enunciado, ID (Integer) de Instancia	hay Ilamado a ningún método de
	de Productcategory, ID (Integer) de	prodRepository (mock), pero hay llamados
	Instancia de Productsubcategory,	a los métodos findById(Integer id) de
	Instancia de Unitmeasure	prodCatRepository (mock) y
		prodSubRepository (mock).
saveProduct	Instancia de Product con	El método lanza NullPointerException con
(Product p,	Sellstartdate=null y el resto de los	mensaje "Cannot invoke
Integer pCatId,	atributos respetando las	"java.sql.Timestamp.getTime()" because
Integer pSubId,	condiciones dadas en el enunciado,	"ts" is null". No hay llamado a ningún
Unitmeasure um)	ID (Integer) de Instancia de	método de prodRepository (mock), pero
	Productcategory, ID (Integer) de	hay llamados a los métodos
	Instancia de Productsubcategory,	findById(Integer id) de prodCatRepository
	Instancia de Unitmeasure	(mock) y prodSubRepository (mock).
saveProduct	Instancia de Product con	El método lanza NullPointerException con
(Product p,	Sellenddate=null y el resto de los	mensaje "Cannot invoke
Integer pCatId,	atributos respetando las	"java.sql.Timestamp.before(java.sql.Timest
Integer pSubId,	condiciones dadas en el enunciado,	amp)" because "Sellenddate" is null". No
Unitmeasure um)	ID (Integer) de Instancia de	hay llamado a ningún método de
Cilitaticasare airi)	in (integer) de instancia de	nay namado a miigan metodo de

	Productcategory, ID (Integer) de	prodRepository (mock), pero hay llamados
	Instancia de Productsubcategory,	a los métodos findById(Integer id) de
	Instancia de Unitmeasure	prodCatRepository (mock) y
		prodSubRepository (mock).
saveProduct	Instancia de Product con los	El método llama correctamente al método
(Product p,	atributos respetando las	save de prodRepository (mock) para
Integer pCatId,	condiciones dadas en el enunciado,	guardar el producto asociado a una
Integer pSubId,	ID (Integer) de Instancia de	categoría, subcategoría y unidad de
Unitmeasure um)	Productcategory, ID (Integer) de	medida y no genera ninguna excepción.
	Instancia de Productsubcategory,	The desired of the general strongers and the general strongers are general strongers and the general strongers and the general strongers are generally and the general strongers and the general strongers and the general strongers and the general strongers are general strongers and the general strongers and the general strongers are generally and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the general strongers and the general strongers are generally as the gener
	Instancia de Unitmeasure	
updateProduct(In	1 (ID de un producto existente),	El método lanza NullPointerException con
teger pld,	null	el mensaje "Cannot invoke
Product p)		"com.taller1.model.prod.Product.getProdu
μ,		ctnumber()" because "p" is null". Hay una
		interacción con prodRepository (mock),
		dado que se hace el llamado al método
		findById(Integer id).
updateProduct(In	1 (ID de un producto existente),	El método lanza Exception con el mensaje
teger pld,	Instancia de Product con	"Product number must be greater than 0".
Product p)	Productnumber menor a 1 y el	Hay una interacción con prodRepository
,	resto de los atributos respetando	(mock), dado que se hace el llamado al
	las condiciones dadas en el	método findById(Integer id).
	enunciado	, , ,
updateProduct(In	1 (ID de un producto existente),	El método lanza Exception con el mensaje
teger pld,	Instancia de Product con	"Product number must be greater than 0".
Product p)	Productnumber="0" y el resto de	Hay una interacción con prodRepository
	los atributos respetando las	(mock), dado que se hace el llamado al
	condiciones dadas en el enunciado	método findByld(Integer id).
updateProduct(In	1 (ID de un producto existente),	El método llama correctamente al método
teger pld,	Instancia de Product con	save de prodRepository (mock) para
Product p)	Productnumber="2" y el resto de	guardar el producto actualizado y no
	los atributos respetando las	genera ninguna excepción.
	condiciones dadas en el enunciado	
updateProduct(In	1 (ID de un producto existente),	El método lanza Exception con el mensaje
teger pld,	Instancia de Product	"Days to manufacture must be greater
Product p)	Daystomanufacture=0 y el resto de	than 0". Hay una interacción con
	los atributos respetando las	prodRepository (mock), dado que se hace
	condiciones dadas en el enunciado	el llamado al método findById(Integer id).
updateProduct(In	1 (ID de un producto existente),	El método llama correctamente al método
teger pld,	Instancia de Product con	save de prodRepository (mock) para
Product p)	Daystomanufacture =2 y el resto de	guardar el producto actualizado y no
	los atributos respetando las	genera ninguna excepción.
	condiciones dadas en el enunciado	
updateProduct(In	1 (ID de un producto existente),	El método lanza NullPointerException con
teger pld,	Instancia de Product con	mensaje "Cannot invoke
Product p)	Sellenddate=null y	"java.sql.Timestamp.before(java.sql.Timest

	T	
	Sellstartdate=null y el resto de los atributos respetando las condiciones dadas en el enunciado	amp)" because "Sellenddate" is null". Hay una interacción con prodRepository (mock), dado que se hace el llamado al método findById(Integer id).
updateProduct(In teger pld, Product p)	1 (ID de un producto existente), Instancia de Product con Sellenddate antes de Sellstartdate y el resto de los atributos respetando las condiciones dadas en el enunciado	El método lanza Exception con mensaje "Sell end date is before sell start date". Hay una interacción con prodRepository (mock), dado que se hace el llamado al método findById(Integer id).
updateProduct(In teger pld, Product p)	1 (ID de un producto existente), Instancia de Product con Sellenddate=null y el resto de los atributos respetando las condiciones dadas en el enunciado	El método lanza NullPointerException con mensaje "Cannot invoke "java.sql.Timestamp.before(java.sql.Timest amp)" because "Sellenddate" is null". Hay una interacción con prodRepository (mock), dado que se hace el llamado al método findById(Integer id).
updateProduct(In teger pld, Product p)	1 (ID de un producto existente), Instancia de Product con Sellstartdate=null y el resto de los atributos respetando las condiciones dadas en el enunciado	El método lanza NullPointerException con mensaje "Cannot invoke "java.sql.Timestamp.getTime()" because "ts" is null". Hay una interacción con prodRepository (mock), dado que se hace el llamado al método findByld(Integer id).
updateProduct(In teger pld, Product p)	1 (ID de un producto existente), Instancia de Product con Sellstartdate antes que Sellenddate y el resto de los atributos respetando las condiciones dadas en el enunciado	El método llama correctamente al método save de prodRepository (mock) para guardar el producto actualizado y no genera ninguna excepción.
PRUEBAS DE INTEG	RACIÓN	
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	null, null, null	El método lanza NullPointerException con mensaje "Producto no puede ser nulo".
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	Instancia de Product, null, ID (Integer) de Instancia de Productsubcategory, Instancia de Unitmeasure	El método lanza NullPointerException con mensaje "No escribió un id de la categoria existente para producto".
saveProduct (Product p, Integer pCatId, Integer pSubId, Unitmeasure um)	Instancia de Product con Productnumber="0" y el resto de los atributos respetando las condiciones dadas en el enunciado, ID (Integer) de Instancia de Productcategory, ID (Integer) de	El método lanza Exception con mensaje "Product number must be greater than 0".

	Instancia de Productsubcategory,	
	Instancia de Productsubcategory,	
saveProduct	Instancia de Ontimeasure	El método lanza Exception con mensaje
		"Sell end date is before sell start date".
(Product p,	Sellenddate antes de Sellstartdate y	Sell ella date is before sell start date .
Integer pCatId,	el resto de los atributos respetando	
Integer pSubId,	las condiciones dadas en el	
Unitmeasure um)	enunciado, ID (Integer) de Instancia	
	de Productcategory, ID (Integer) de	
	Instancia de Productsubcategory,	
	Instancia de Unitmeasure	
saveProduct	Instancia de Product con los	El método se ejecuta correctamente, llama
(Product p,	atributos respetando las	al método save de ProductRepository y
Integer pCatId,	condiciones dadas en el enunciado,	queda registrado dentro de la base de
Integer pSubId,	ID (Integer) de Instancia de	datos del CrudRepository (Se verifica con el
Unitmeasure um)	Productcategory, ID (Integer) de	método findById del repositorio y
	Instancia de Productsubcategory,	comparando el Productid y Sellstartdate
	Instancia de Unitmeasure	del encontrado con el insertado)
updateProduct(In	Instancia inicial de Product con	El método lanza NullPointerException con
teger pld,	Productnumber="1",	mensaje "Cannot invoke
Product p)	Daystomanufacture=1,	"com.taller1.model.prod.Product.getProdu
	Sellstartdate=new Timestamp	ctnumber()" because "p" is null".
	(System.currentTimeMillis()-100),	
	Sellenddate=new Timestamp	
	(System.currentTimeMillis()-100),	
	Productid=1, name="Name	
	Product", color="Blue",	
	size="Small"	
	Book water	
	Parámetros:	
1 . 5	1, null	
updateProduct(In	Instancia inicial de Product con	El método se ejecuta correctamente, llama
teger pld,	Productnumber="1",	al método findByld y save de
Product p)	Daystomanufacture=1,	ProductRepository y el producto queda
	Sellstartdate=new Timestamp	actualizado dentro de la base de datos del
	(System.currentTimeMillis()-100),	CrudRepository (Se verifica comparando el
	Sellenddate=new Timestamp	name del producto insertado, con el name
	(System.currentTimeMillis()-100),	del producto encontrado)
	Productid=1, name="Name	
	Product", color="Blue",	
	size="Small"	
	Darámetros	
	Parámetros:	
	1, Instancia de Product con los	
	atributos de la instancia inicial pero	
	con name="New name"	

updateProduct(In teger pld, Product p)	Instancia inicial de Product con Productnumber="1", Daystomanufacture=1, Sellstartdate=new Timestamp (System.currentTimeMillis()-100), Sellenddate=new Timestamp (System.currentTimeMillis()-100), Productid=1, name="Name Product", color="Blue", size="Small"	El método se ejecuta correctamente, llama al método findByld y save de ProductRepository y el producto queda actualizado dentro de la base de datos del CrudRepository (Se verifica comparando el color del producto insertado, con el color del producto encontrado)
	Parámetros: 1, Instancia de Product con los atributos de la instancia inicial pero con color="Red"	
updateProduct(In teger pld, Product p)	Instancia inicial de Product con Productnumber="1", Daystomanufacture=1, Sellstartdate=new Timestamp (System.currentTimeMillis()-100), Sellenddate=new Timestamp (System.currentTimeMillis()-100), Productid=1, name="Name Product", color="Blue", size="Small" Parámetros:	El método se ejecuta correctamente, llama al método findByld y save de ProductRepository y el producto queda actualizado dentro de la base de datos del CrudRepository (Se verifica comparando el size del producto insertado, con el size del producto encontrado)
	1, Instancia de Product con los atributos de la instancia inicial pero con size="Medium"	
updateProduct(In teger pld, Product p)	Instancia inicial de Product con Productnumber="1", Daystomanufacture=1, Sellstartdate=new Timestamp (System.currentTimeMillis()-100), Sellenddate=new Timestamp (System.currentTimeMillis()-100), Productid=1, name="Name Product", color="Blue", size="Small"	El método se ejecuta correctamente, llama al método findByld y save de ProductRepository y el producto queda actualizado dentro de la base de datos del CrudRepository (Se verifica comparando el Productnumber del producto insertado, con el Productnumber del producto encontrado)
	Parámetros: 1, Instancia de Product con los atributos de la instancia inicial pero con Productnumber="2"	
updateProduct(In teger pld, Product p)	Instancia inicial de Product con Productnumber="1", Daystomanufacture=1,	El método se ejecuta correctamente, llama al método findByld y save de ProductRepository y el producto queda

Sellstartdate=new Timestamp	actualizado dentro de la base de datos del
(System.currentTimeMillis()-100),	CrudRepository (Se verifica comparando el
Sellenddate=new Timestamp	Sellstartdate del producto insertado, con el
(System.currentTimeMillis()-100),	Sellstartdate del producto encontrado)
Productid=1, name="Name	
Product", color="Blue",	
size="Small"	
Parámetros:	
1, Instancia de Product con los	
atributos de la instancia inicial pero	
con Sellenddate=new	
Timestamp(System.currentTimeMill	
is()+1000) y Sellstartdate(new	
Timestamp(System.currentTimeMill	
is()-1000)	

Pruebas sobre la implementación de com.taller1.service.implementation.SpecialOfferServiceImpl PRUEBAS UNITARIAS Método a probar Paramétro(s) Resultado esperado (Prueba unitaria) saveSpecialOffer(Spe null El método lanza NullPointerException cialoffer s) con mensaje "SpecialOffer cannot be null". No hay llamado a ningún método de soRepository (mock). saveSpecialOffer(Spe Instancia de Specialoffer con El método lanza Exception con cialoffer s) category=null y el resto de los mensaje "SpecialOffer's category atributos respetando las cannot be null". No hay llamado a condiciones dadas en el enunciado. ningún método de soRepository (mock). saveSpecialOffer(Spe Instancia de Specialoffer con El método lanza Exception con cialoffer s) mensaje "SpecialOffer's discount must discountpct=null y el resto de los atributos respetando las be greater than 0". No hay llamado a condiciones dadas en el enunciado. ningún método de soRepository (mock). El método lanza Exception con saveSpecialOffer(Spe Instancia de Specialoffer con cialoffer s) discountpct=BigDecimal.valueOf(-1) mensaje "SpecialOffer's discount must y el resto de los atributos be greater than 0". No hay llamado a respetando las condiciones dadas ningún método de soRepository (mock). en el enunciado. saveSpecialOffer(Spe Instancia de Specialoffer con El método lanza Exception con mensaje "SpecialOffer's modified date cialoffer s) modifieddate=null y el resto de los atributos respetando las must be now and cannot be null". No condiciones dadas en el enunciado. hay llamado a ningún método de soRepository (mock).

saveSpecialOffer(Spe cialoffer s) saveSpecialOffer(Spe cialoffer s)	Instancia de Specialoffer con modifieddate=new Timestamp(System.currentTimeMill is()+100000) y el resto de los atributos respetando las condiciones dadas en el enunciado. Instancia de Specialoffer con los atributos respetando las	El método lanza Exception con mensaje "SpecialOffer's modified date must be now and cannot be null". No hay llamado a ningún método de soRepo (mock). El método llama correctamente al método save de soRepo (mock) para
updateSpecialOffer(I nteger sld, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMill	guardar la Specialoffer y no genera ninguna excepción. El método lanza NoSuchElementException con el mensaje "no value present". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findByld(Integer id).
updateSpecialOffer(I	is()-100) enddate= new Timestamp(System.currentTimeMill is()+1000) Parametros: 0, Instancia inicial de Specialoffer Instancia inicial de Specialoffer con category="Special Category"	El método lanza NullPointerException
nteger sld, Specialoffer s)	category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMill is()-100) enddate= new Timestamp(System.currentTimeMill is()+1000) Parametros: 1, null	con el mensaje "Cannot invoke "com.taller1.model.sales.Specialoffer.g etCategory()" because "s" is null". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findById(Integer id).
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new	El método lanza Exception con el mensaje "Category cannot be null or blank". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findById(Integer id).

	Timestamp(System.currentTimeMill is()-100) enddate= new Timestamp(System.currentTimeMill is()+1000) Parametros: 1, Instancia de Specialoffer con los atributos de la instancia inicial pero	
updateSpecialOffer(I	con category=null Instancia inicial de Specialoffer con	El método lanza Exception con el
nteger sld, Specialoffer s)	category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMill is()-100) enddate= new Timestamp(System.currentTimeMill is()+1000)	mensaje "Discount cannot be null nor less than 0". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findById(Integer id).
	Parametros: 1, Instancia de Specialoffer con los atributos de la instancia inicial pero con discountpct=null	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()+1000)	El método lanza Exception con el mensaje "Discount cannot be null nor less than 0". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findById(Integer id).
	Parametros: 1, Instancia de Specialoffer con los atributos de la instancia inicial pero con discountpct=BigDecimal.valueOf(-1)	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp	El método lanza Exception con el mensaje "Discount cannot be null nor less than 0". Hay una interacción con soRepo (mock), dado que se hace el

	(System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMill is()-100) enddate= new Timestamp(System.currentTimeMill is()+1000) Parametros:	llamado al método findById(Integer id).
	1, Instancia de Specialoffer con los atributos de la instancia inicial pero con discountpct=BigDecimal.valueOf(0)	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()+1000)	El método lanza Exception con el mensaje "Modified date must be now and not null". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findById(Integer id).
	Parametros: 1, Instancia de Specialoffer con los atributos de la instancia inicial pero con modifieddate=null	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()+1000)	El método lanza Exception con el mensaje "Modified date must be now and not null". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findById(Integer id).
	Parametros: 1, Instancia de Specialoffer con los atributos de la instancia inicial pero con modifieddate=new	

	Timestamp(System.currentTimeMill	
updateSpecialOffer(I nteger sId, Specialoffer s)	is()-100000) Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()-100)	El método lanza Exception con el mensaje "Modified date must be now and not null". Hay una interacción con soRepo (mock), dado que se hace el llamado al método findById(Integer id).
	is()+1000) Parametros: 1, Instancia de Specialoffer con los atributos de la instancia inicial pero con modifieddate=new Timestamp(System.currentTimeMill is()+100000)	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()+1000) Parametros:	El método llama correctamente al método save de soRepo (mock) para guardar el producto actualizado y no genera ninguna excepción.
	1, Instancia de Specialoffer con los atributos de la instancia inicial y maxqty=15 y minqty=5	
PRUEBAS DE INTEGRAC	CIÓN	
saveSpecialOffer(Spe cialoffer s)	null	El método lanza NullPointerException con mensaje "SpecialOffer cannot be null"
saveSpecialOffer(Spe cialoffer s)	Instancia de Specialoffer con category=null y el resto de los atributos respetando las condiciones dadas en el enunciado	El método lanza NullPointerException con mensaje "SpecialOffer's category cannot be null"

saveSpecialOffer(Spe	Instancia de Specialoffer con	El método lanza NullPointerException
cialoffer s)	discountpct=BigDecimal.valueOf(-1)	con mensaje "SpecialOffer's discount
	y el resto de los atributos	must be greater than 0"
	respetando las condiciones dadas	J
	en el enunciado	
saveSpecialOffer(Spe	Instancia de Specialoffer con	El método lanza NullPointerException
cialoffer s)	modifieddate=new	con mensaje "SpecialOffer's modified
,	Timestamp(System.currentTimeMill	date must be now and cannot be null"
	is()-100000) y el resto de los	
	atributos respetando las	
	condiciones dadas en el enunciado	
saveSpecialOffer(Spe	Instancia de Specialoffer con los	El método se ejecuta correctamente,
cialoffer s)	atributos respetando las	llama al método save de
	condiciones dadas en el enunciado	SpecialofferRepository y queda
		registrado dentro de la base de datos
		del CrudRepository (Se verifica con el
		método findById del repositorio y
		comparando el Specialofferid y
		description del Specialoffer
		encontrado con el insertado)
updateSpecialOffer(I	Instancia inicial de Specialoffer con	El método lanza
nteger sld,	category="Special Category",	NoSuchElementException con el
Specialoffer s)	discountpct=BigDecimal.valueOf(5),	mensaje "no value present".
	modifieddate=new Timestamp	
	(System.currentTimeMillis()),	
	specialofferid=1,	
	description="Description",	
	startdate= new	
	Timestamp(System.currentTimeMill	
	is()-100) enddate= new	
	Timestamp(System.currentTimeMill	
	is()+1000)	
	Parámetros	
	Parámetros: 0, Instancia de Specialoffer inicial	
updateSpecialOffer(I	Instancia inicial de Specialoffer con	El método lanza NullPointerException
nteger sld,	category="Special Category",	con el mensaje "Cannot invoke
Specialoffer s)	discountpct=BigDecimal.valueOf(5),	"com.taller1.model.sales.Specialoffer.g
Specialoffer 3)	modifieddate=new Timestamp	etCategory()" because "s" is null".
	(System.currentTimeMillis()),	
	specialofferid=1,	
	description="Description",	
	startdate= new	
	Timestamp(System.currentTimeMill	
	is()-100) enddate= new	
	Timestamp(System.currentTimeMill	
	is()+1000)	

	Parámetros:	
	1, null	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()+1000)	El método lanza Exception con el mensaje "Category cannot be null or blank".
	Parámetros: 1, Instancia inicial de Specialoffer con category=null	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()+1000) Parámetros:	El método lanza Exception con el mensaje "Discount cannot be null nor less than 0".
	1, Instancia inicial de Specialoffer con discountpct=BigDecimal.valueOf(0)	
updateSpecialOffer(I nteger sId, Specialoffer s)	Instancia inicial de Specialoffer con category="Special Category", discountpct=BigDecimal.valueOf(5), modifieddate=new Timestamp (System.currentTimeMillis()), specialofferid=1, description="Description", startdate= new Timestamp(System.currentTimeMillis()-100) enddate= new Timestamp(System.currentTimeMillis()+1000)	El método se ejecuta correctamente, llama al método findByld y save de SpecialofferRepository y queda registrado dentro de la base de datos del CrudRepository (Se verifica con el método findByld del repositorio y comparando la description del Specialoffer encontrado con el insertado)

Parametros:	
1, Instancia de Specialoffer con los	
atributos de la instancia inicial y	
maxqty=15 y minqty=5	

Pruebas sobre la implementación de com.taller1.service.implementation.SpecialOfferProductServiceImpl

PRUEBAS UNITARIAS

Método a probar	Paramétro(s)	Resultado esperado (Prueba unitaria)
saveSpecialOfferProduct(Sp ecialofferproduct sp, Integer pld, Integer sold)	null, 1, 1	El método lanza NullPointerException con mensaje "SpecialOfferProduct must not be null". No hay llamado a ningún método de sopRep (mock).
saveSpecialOfferProduct(Sp ecialofferproduct sp, Integer pld, Integer sold)	Instancia de Specialofferproduct con los atributos respetando las condiciones dadas en el enunciado, 0, 1	El método lanza NoSuchElementException con mensaje "no value present". No hay llamado a ningún método de sopRep (mock), pero si hay un llamado al método findByld(Integer id) de pRep (mock).
saveSpecialOfferProduct(Sp ecialofferproduct sp, Integer pld, Integer sold)	Instancia de Specialofferproduct con los atributos respetando las condiciones dadas en el enunciado, 1, 0	El método lanza NoSuchElementException con mensaje "no value present". No hay llamado a ningún método de sopRep (mock), pero si hay un llamado al método findByld(Integer id) de pRep (mock) y soRep (mock).
saveSpecialOfferProduct(Sp ecialofferproduct sp, Integer pld, Integer sold)	Instancia de Specialofferproduct con modifieddate=null y el resto de los atributos respetando las condiciones dadas en el enunciado, 1, 1	El método lanza Exception con mensaje "SpecialOfferProduct modified date must be now and cannot be null". No hay llamado a ningún método de sopRep (mock).
saveSpecialOfferProduct(Sp ecialofferproduct sp, Integer pld, Integer sold)	Instancia de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()-100000)y el resto de los atributos respetando las condiciones dadas en el enunciado, 1, 1	El método lanza Exception con mensaje "SpecialOfferProduct modified date must be now and cannot be null". No hay llamado a ningún método de sopRep (mock).
saveSpecialOfferProduct(Sp ecialofferproduct sp, Integer pld, Integer sold)	Instancia de Specialofferproduct con modifieddate=new Timestamp(System.currentTi	El método lanza Exception con mensaje "SpecialOfferProduct modified date must be now and

	meMillis()+100000)y el resto de los atributos respetando las condiciones dadas en el enunciado, 1, 1	cannot be null". No hay llamado a ningún método de sopRep (mock).
saveSpecialOfferProduct(Sp ecialofferproduct sp, Integer pld, Integer sold)	Instancia de Specialofferproduct con los atributos respetando las condiciones dadas en el enunciado, 1, 1	El método llama correctamente al método save de sopRepo (mock) para guardar la Specialofferproduct y no genera ninguna excepción.
editSpecialOfferProduct(Sp ecialofferproductPK spld, Specialofferproduct sp)	Instancia inicial de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()), id=Instancia de SpecialofferproductPK Parámetros: Intancia de	El método lanza NullPointerException con mensaje "Cannot invoke "com.taller1.model.sales.Specialofferp roduct.getModifieddate()" because "sp" is null".
	SpecialofferproductPK, null	
editSpecialOfferProduct(Sp ecialofferproductPK spId, Specialofferproduct sp)	Instancia inicial de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()), id=Instancia de SpecialofferproductPK Parámetros: Intancia de SpecialofferproductPK,	El método lanza Exception con mensaje "Modified date must be now and not null".
	Instancia inicial de Specialofferproduct con	
line is less as the sta	modifieddate=null	
editSpecialOfferProduct(Sp ecialofferproductPK spld, Specialofferproduct sp)	Instancia inicial de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()), id=Instancia de SpecialofferproductPK	El método lanza Exception con mensaje "Modified date must be now and not null".
	Parámetros: Intancia de SpecialofferproductPK, Instancia inicial de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()-100000)	

editSpecialOfferProduct(Sp ecialofferproductPK spld, Specialofferproduct sp)	Instancia inicial de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()), id=Instancia de SpecialofferproductPK	El método lanza Exception con mensaje "Modified date must be now and not null".
	Parámetros: Intancia de SpecialofferproductPK, Instancia inicial de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()+100000)	
editSpecialOfferProduct(Sp ecialofferproductPK spId, Specialofferproduct sp)	Instancia inicial de Specialofferproduct con modifieddate=new Timestamp(System.currentTi meMillis()), id=Instancia de SpecialofferproductPK Parámetros: Intancia de SpecialofferproductPK, Instancia inicial de Specialofferproduct con rowguid=2	El método llama correctamente al método save de sopRep (mock) para guardar el specialofferproduct actualizado y no genera ninguna excepción.
PRUEBAS DE INTEGRACIÓN		