



UNIVERSIDAD DE LA FRONTERA
FACULTAD DE INGENIERÍA Y CIENCIAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

SOFTWARE PARA ADMINISTRAR STOCK

PROFESOR : Juan Lagos Obando

AUTOR: Diego Aido Subiabre, Vicente Hernández
Temuco, 24 de Septiembre 2025

Índice general

1. Introducción	1
1.1. Descripción del problema	2
1.2. Propuesta de solución	2
1.3. Diferenciador con soluciones existente	3
2. Diagramas	5
3. Repositorio	9

Índice de figuras

2.1. Diagrama de clases [UML]	6
2.2. Diagrama de Casos de uso	7
2.3. BPMN a nivel de procesos del caso	7
2.4. BPMN a nivel de actividades (Vicente Hernández)	8
2.5. BPMN a nivel de actividades (Diego Aido)	8
3.1. Rúbrica de Evaluación	10

Capítulo 1

Introducción

1. Introducción

1.1. Descripción del problema

Se busca hacer un programa el cual pueda funcionar para supermercados o tiendas pequeñas las cuales puedan optar por tecnología buenas sin gastar dinero, dado que se está comenzando. Sin embargo, lo que se busca es llegar a satisfacer los siguientes puntos que son esenciales:

Administrar stock: En este punto lo que se necesita poder hacer que un usuario de tipo administrador, tenga la capacidad administrar los productos de alguna tienda, y de alguna forma hacer que se de forma fácil de usar o entender, rápido y seguro

Simular venta: Lo que se necesita llevar a cabo que un usuario pueda efectuar una compra en alguna caja que puede existir.

Visualizar a futuro: Se necesita que sistema que detecte cuando algún producto posiblemente esté casi agotado, guardar datos de lo que más se vende en algún tiempo, lo que más se vende, entre otras más.

1.2. Propuesta de solución

Administrar stock: En este punto lo que se busca es poder hacer que un usuario de tipo administrador, pueda administrar los productos, donde el puede saber que productos hay dentro de la tienda en los “estantes”, y en las “bodegas”, también es muy importante poder identificar los estantes, bodegas y productos, también hacer una diferenciación por categoría, etc.

Simular venta: Lo que se busca llevar a cabo es poder hacer que un usuario pueda efectuar una compra en alguna caja que puede existir, donde los posibles métodos de pagos son con tarjeta de débito o crédito o simplemente efectivo, entregarle una boleta donde se detalla lo de una boleta

Visualizar a futuro: Lo que se espera es poder hacer un sistema que detecte cuando un producto está por agotarse, qué es lo más vendido en este tiempo, poder registrar lo más vendido por temporada, etc. Para poder tomar decisiones que sean un beneficio financiero a futuro.

1.3. Diferenciador con soluciones existente

En el mercado actual, existen diversas soluciones para la gestión de stock en pequeñas tiendas y supermercados, tanto gratuitas como de pago. Algunas de las más populares incluyen Odoo, Loyverse TPV y Square Inventory. A continuación, se detalla cómo nuestra propuesta se diferencia de estas alternativas, enfocándonos en aspectos como el costo, la simplicidad, las funcionalidades específicas y la adaptabilidad para negocios emergentes en contextos locales como Chile.

Comparación con Odoo: Odoo es un software ERP de código abierto con una versión gratuita que permite gestionar inventarios, ventas y reportes. Sin embargo, su versión comunitaria tiene limitaciones en funcionalidades avanzadas (como integraciones premium o soporte técnico), y requiere una curva de aprendizaje pronunciada debido a su complejidad modular. Nuestra solución, en cambio, es completamente gratuita sin restricciones ocultas, diseñada específicamente para usuarios principiantes con una interfaz intuitiva y rápida. Además, incorpora diferenciación detallada entre "estantes" y "bodegas", algo que en Odoo requiere configuración adicional, y enfatiza predicciones futuras (como detección de stock bajo o análisis estacional) de manera integrada y sencilla, sin necesidad de módulos extras.

Comparación con Loyverse TPV: Loyverse es un sistema de punto de venta móvil gratuito, ideal para pequeñas tiendas, que maneja ventas, inventarios y pagos (incluyendo tarjetas y efectivo). Ofrece sincronización en la nube y reportes básicos, pero depende de internet para funcionalidades completas y puede tener costos por hardware o características premium (como lealtad de clientes). Nuestra propuesta se distingue por ser independiente de conexiones en línea en su núcleo (permitiendo operación offline en entornos con conectividad limitada, común en regiones rurales de Chile), y por incluir visualización predictiva avanzada, como registro de productos más vendidos por temporada, lo cual no es nativo en la versión gratuita de Loyverse. Además, generamos boletas detalladas de forma automática y segura, priorizando la privacidad de datos sin reliance en servicios externos.

Comparación con Square Inventory: Square proporciona un software de gestión de inventario gratuito integrado con su punto de venta, enfocado en tracking de stock y alertas de bajo inventario. Es accesible para pequeños negocios, pero está orientado a ecosistemas con hardware propio (como lectores de tarjetas), y sus análisis predictivos son limitados en la versión base, con énfasis en ventas en tiempo real. Lo que nos diferencia es el enfoque en la personalización para tiendas locales, como categorización por estantes y bodegas, y herramientas de decisión financiera a futuro (e.g., identificación de tendencias de ventas) sin costos adicionales. Nuestra solución es open-source (disponible en GitHub), permitiendo modificaciones por la comunidad, lo cual fomenta la adaptabilidad para necesidades específicas de supermercados pequeños que inician sin

presupuesto, a diferencia de Square, que es más cerrado y orientado a escalabilidad pagada.

En resumen, mientras las soluciones existentes son robustas, a menudo implican complejidad, dependencias externas o limitaciones en versiones gratuitas. Nuestra propuesta destaca por su gratuidad absoluta, simplicidad de uso, seguridad inherente y funcionalidades predictivas orientadas a la toma de decisiones financieras, haciendo que sea ideal para tiendas emergentes que buscan tecnología accesible sin compromisos.

Capítulo 2

Diagramas

2. Diagramas

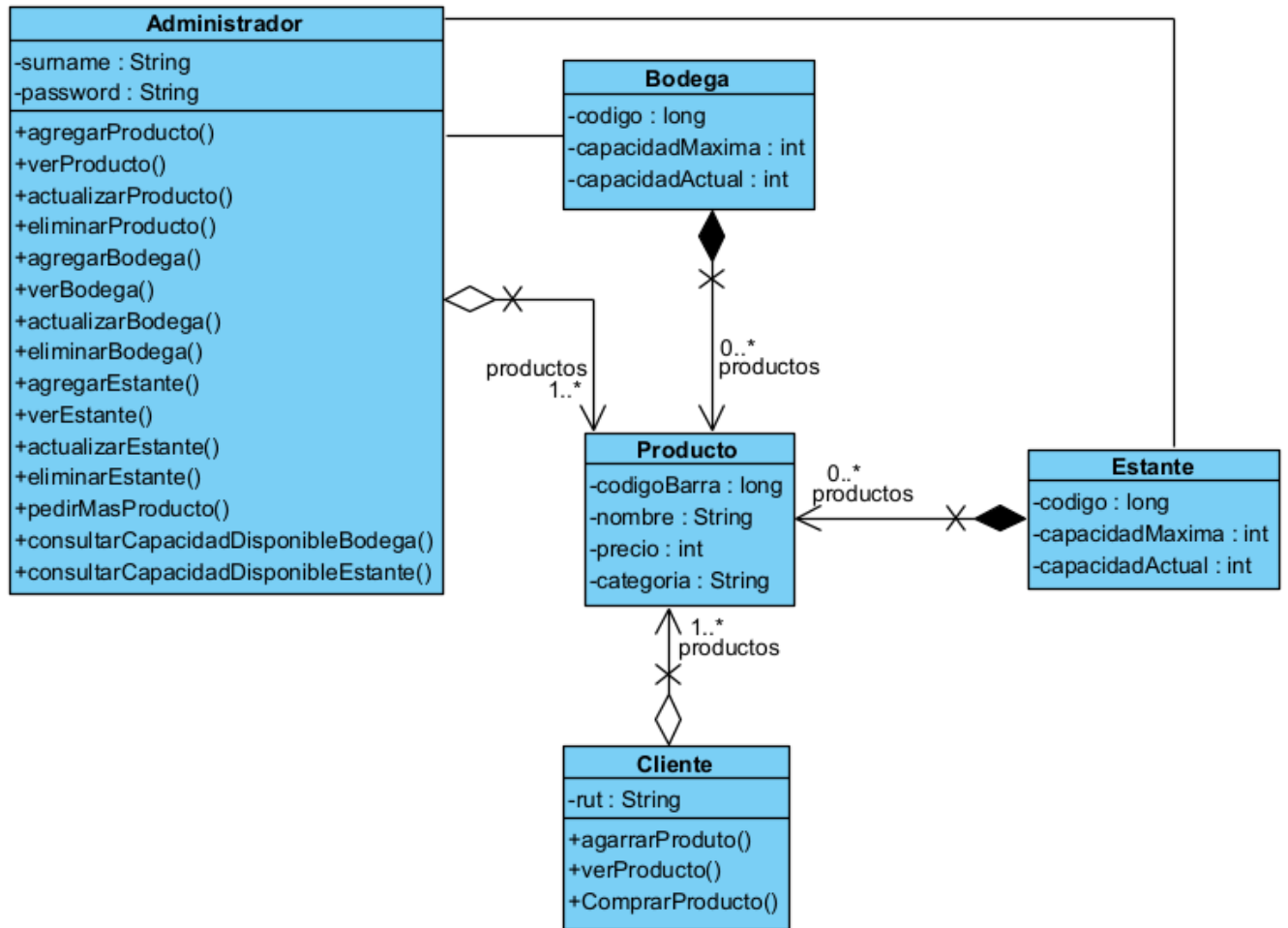


Figura 2.1: Diagrama de clases [UML]

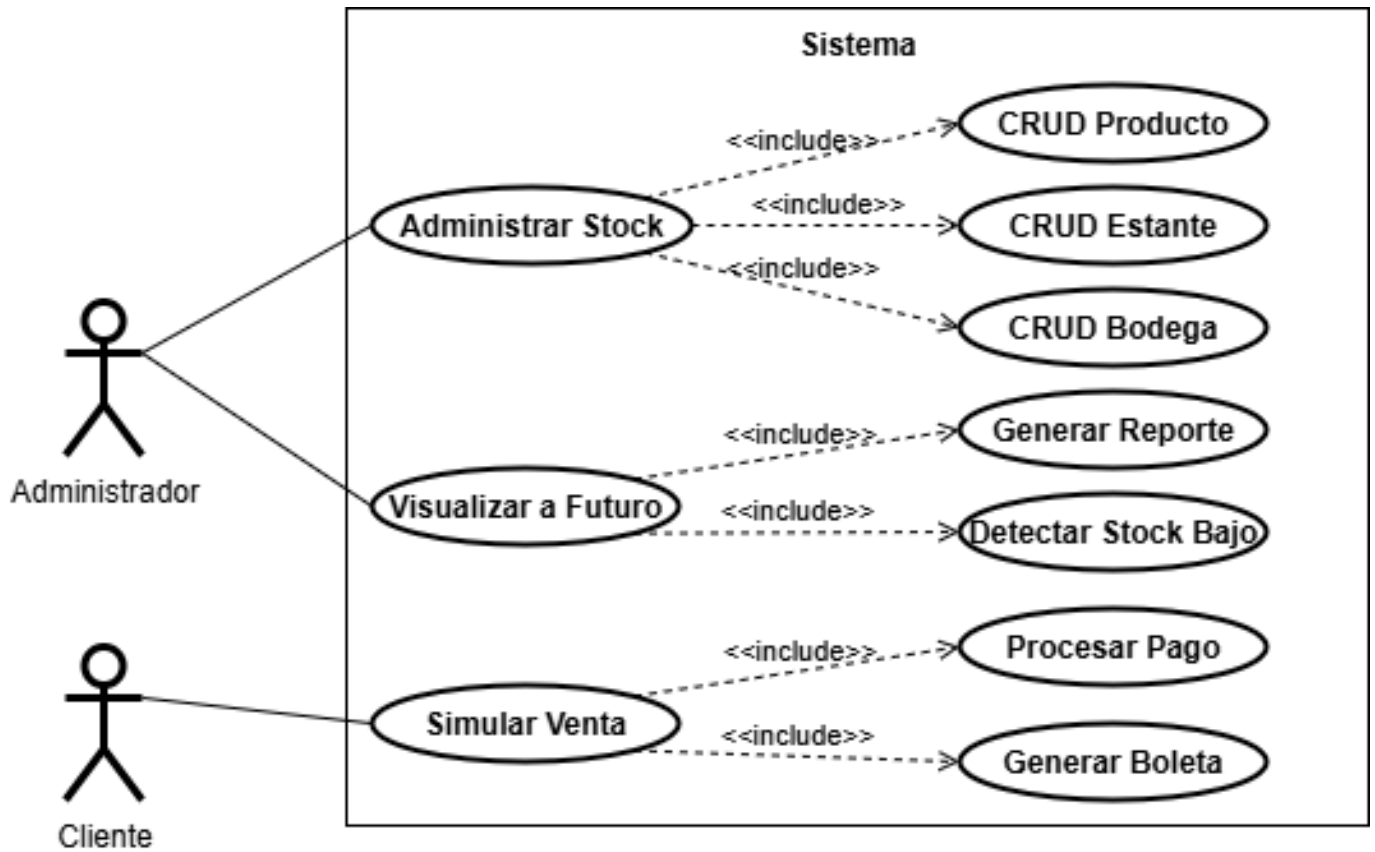


Figura 2.2: Diagrama de Casos de uso

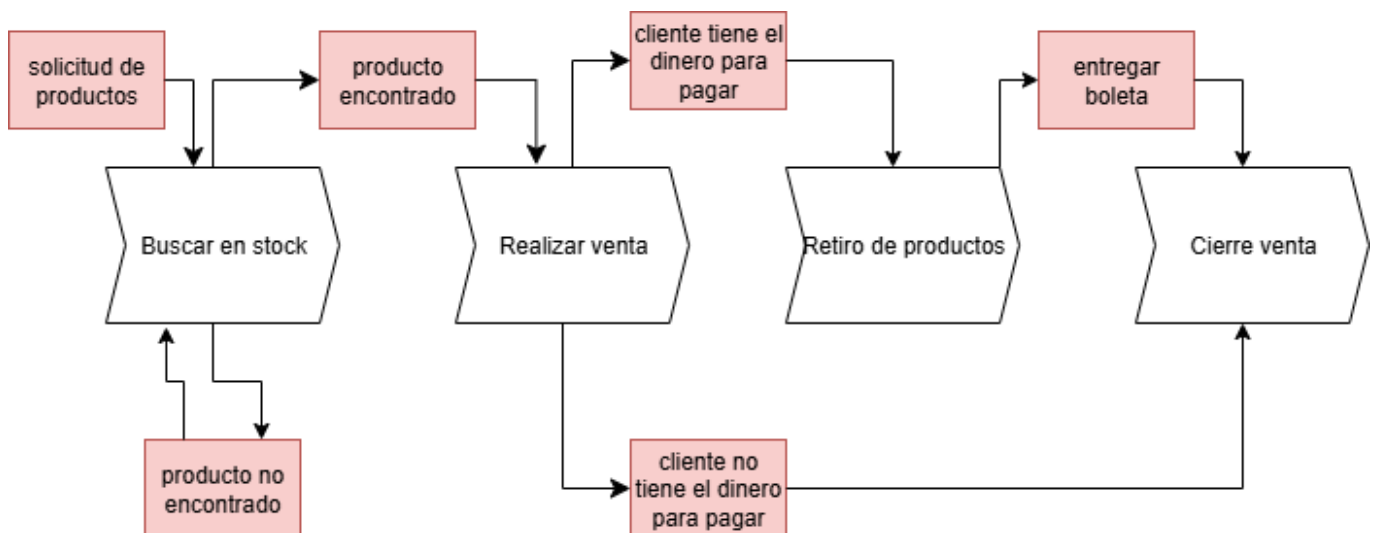


Figura 2.3: BPMN a nivel de procesos del caso

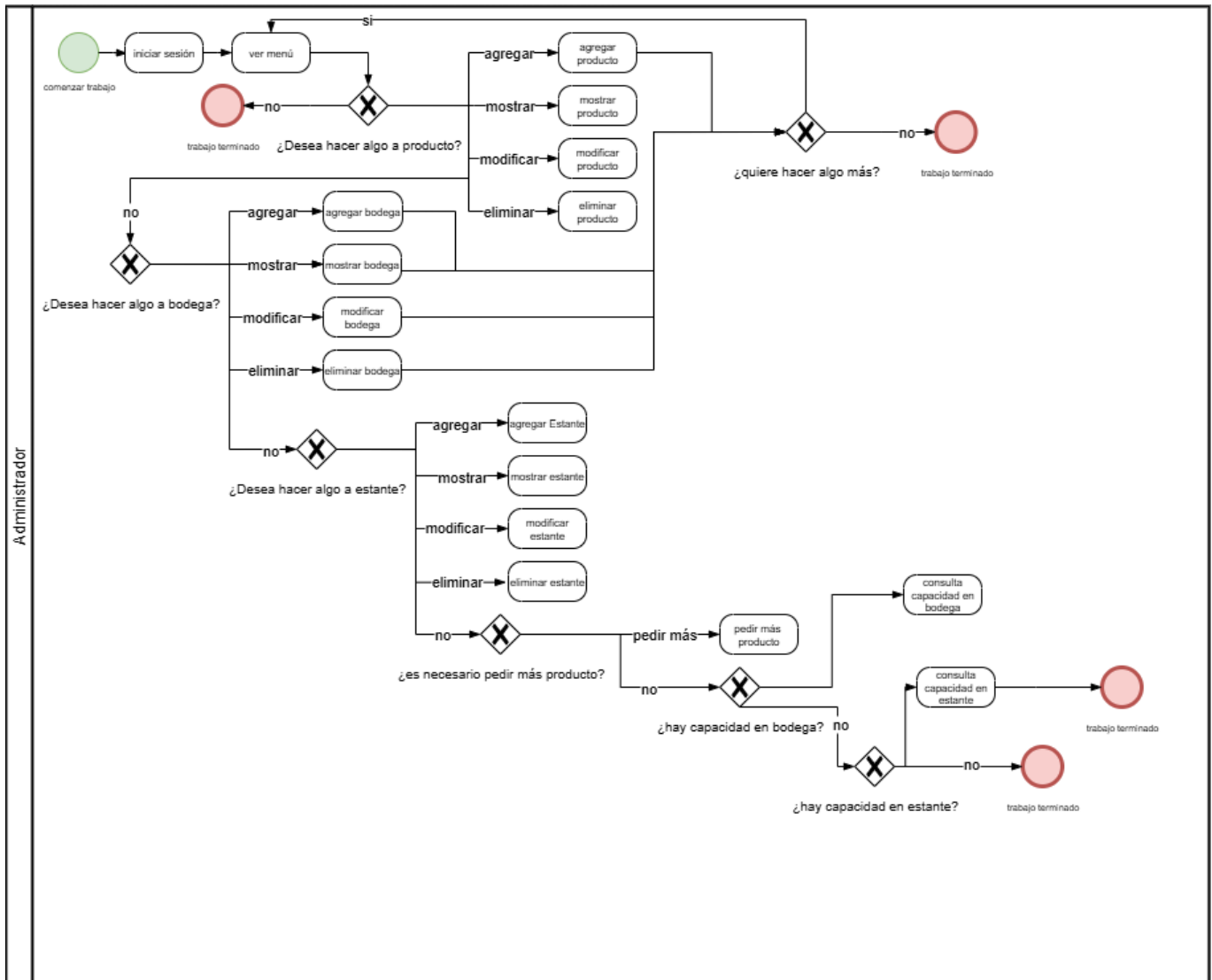


Figura 2.4: BPMN a nivel de actividades (Vicente Hernández)

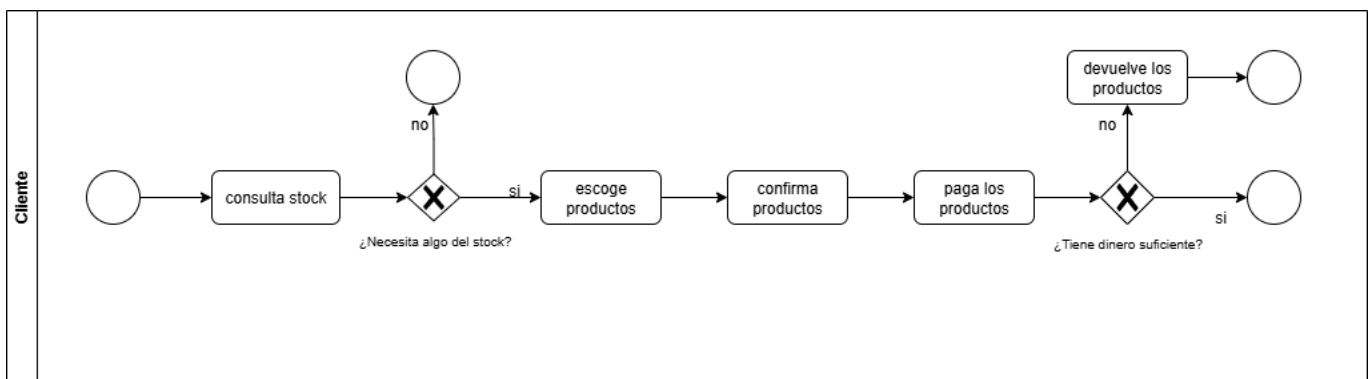


Figura 2.5: BPMN a nivel de actividades (Diego Aido)

Capítulo 3

Repositorio

3. Repositorio

github.com

Criterio	4 puntos	2 puntos	1 punto	0 puntos
Entregables (informe PDF y link público al repositorio)	Entrega ambos: informe en PDF y repositorio accesible.	Entrega solo uno de los dos.	Entrega uno, pero con defecto (informe no es PDF o repositorio no accesible).	No entrega ninguno.
Descripción del caso	El caso tiene coherencia y sentido como producto logable. Se destaca el análisis del alcance diferenciando lo obligatorio, lo ideal y lo poco probable.	El caso es coherente pero el análisis de alcance es incompleto o poco claro.	El caso tiene debilidades en coherencia y el análisis de alcance es mínimo.	No presenta el caso o es ininteligible.
Diagrama de clases (abstracción)	Diagramas claros y exportados como imagen; abstracción de clases totalmente correcta.	Diagramas entregados, con errores menores de abstracción.	Diagramas con errores importantes de abstracción.	No entrega diagrama o es ininteligible.
Diagrama de clases (asociaciones)	Todas las asociaciones correctas.	Hasta 2 asociaciones con defectos evidentes.	Hasta 4 asociaciones con defectos evidentes.	Más de 4 asociaciones con defectos evidentes o no entrega.
Casos de uso (componentes)	Sistema, actores y casos de uso bien identificados y representados.	Los 3 componentes existen pero con defectos de identificación o representación.	Falta 1 componente del diagrama.	Faltan 2 o más componentes.
Casos de uso (relaciones)	Todas las relaciones correctas.	Hasta 2 relaciones con defectos evidentes.	Hasta 4 relaciones con defectos evidentes.	Más de 4 relaciones con defectos evidentes o no entrega.
BPMN a nivel de procesos	Modelo claro, correcto y bien estructurado a nivel de procesos.	Modelo correcto en lo general, pero con errores menores de notación o estructura.	Modelo con varios errores o incompleto.	No presenta modelo o es disfuncional.
BPMN a nivel de actividades	Modelo claro, correcto y bien estructurado a nivel de actividades.	Modelo correcto en lo general, pero con errores menores de notación o estructura.	Modelo con varios errores o incompleto.	No presenta modelo o es disfuncional.

Figura 3.1: Rúbrica de Evaluación