

# UNIVERSIDAD MAYOR, REAL Y PONTIFICIA DE SAN FRANCISCO XAVIER DE CHUQUISACA FACULTAD DE CIENCIAS Y TECNOLOGÍA



## Documentación 'Mi-Tienda'

**Materia:** Calidad de Software (COM450)

**Universitarios:**

- Yamil Fernando Torrez García (CU: 111-324)
- Diego Armando Párraga Ortuste (CU: 111-102)

**Carrera:** Ing. Ciencias de la Computación

**Docente:** Ing. Montellano Barriga Carlos David

SUCRE-BOLIVIA

# 1 Contenido

1.	5	
2.	6	
3.	8	
	Diagrama de casos de uso	8
	Diagrama de clases	9
	Diagrama entidad Relación	10
	Diagrama de componentes	11
	Diagrama de Despliegue	11
4.	12	
5.	12	
5.1	12	
5.2	12	
5.3	13	
6.	13	
6.1	13	
6.1.1	13	
6.1.2	13	
6.1.3	13	
6.1.4	13	
6.2	13	
6.2.1	13	
6.2.2	14	
6.2.3	14	
6.3	14	
6.3.1	14	
6.3.2	14	
6.3.3	14	
7	14	
7.1	14	
7.1.1	14	
7.1.2	14	
7.1.3	14	
7.2	14	

7.2.1	14
7.2.2	14
7.2.3	14
7.3	15
7.3.1	15
7.3.2	15
7.3.3	15
8	15
8.1	15
8.1.1	15
8.1.2	15
8.1.3	15
9	15
9.1	15
9.1.1	15
9.1.2	15
9.2	15
9.2.1	15
9.2.2	16
9.3	16
9.3.1	16
9.3.2	16
10	16
10.1.1	16
10.1.1.1	16
10.1.1.2	16
10.1.2	16
10.1.2.1	17
10.1.2.2	17
10.1.2.3	17
10.1.3	17
10.1.3.1	17
10.1.3.2	17
10.1.3.3	17
11	17
11.1	17

**11.1.1** 17

**11.1.2** 17

**11.1.3** 17

# Documentación del software para hacer la evaluación



## 1.Introducción

La documentación del software es una tarea esencial para cualquier proyecto de desarrollo de software, ya que proporciona información detallada sobre el funcionamiento del sistema.

En este proyecto, se va a documentar un software de una tienda web desarrollado en Laravel. Laravel es un framework de desarrollo de aplicaciones web en PHP que permite crear aplicaciones robustas y escalables.

La tienda web es un sistema complejo que permite la compra y venta de productos en línea, por lo que es importante documentar su arquitectura, funcionalidades y proceso de desarrollo para asegurar su mantenimiento y evolución a largo plazo.

A través de esta documentación, se espera proporcionar una guía completa y detallada sobre el software de la tienda web en Laravel, para que los desarrolladores y los usuarios puedan entender el funcionamiento del sistema y hacer mejoras y modificaciones necesarias.

Nosotros empleamos el software situado en el siguiente repositorio:

<https://github.com/El-Posho-Web/Mi-Tienda>

‘Mi-Tienda’: es una aplicación web desarrollada por alumnos de la carrera de Ingeniería en Informática de la Universidad Católica de Santiago del Estero para la materia de Auditoria en Informática utilizando el framework de Laravel.

## 2.Requisitos para la instalación

### Requerimientos

- [Laravel](#)
- [NodeJs](#)
- [Composer](#)
- [Php](#)
- [Mysql](#)

Opcional para Php y Mysql

- [XAMPP](#)

### Instalación

Para poder correr el repositorio localmente:

1. Clonar el repositorio  
`git clone https://github.com/El-Posho-Web/Mi-Tienda.git`
2. Ir a la raiz del directorio  
`cd Mi-Tienda`
3. Copiar el archivo .env.example a .env  
`cp .env.example .env`
4. Crear la base de datos mi-tienda (puedes cambiar el nombre de la base de datos)
5. Go to .env file
  - Setear las credenciales de la base de datos (DB\_DATABASE=mi-tienda, DB\_USERNAME=root, DB\_PASSWORD=)
6. Instalar las dependencias de PHP  
`composer install`
7. Generar la key  
`php artisan key:generate`
8. Instalar las dependencias de Front-End  
`npm install && npm run dev`
9. Correr las migraciones  
`php artisan migrate`

10. Correr el seeder

`php artisan db:seed`

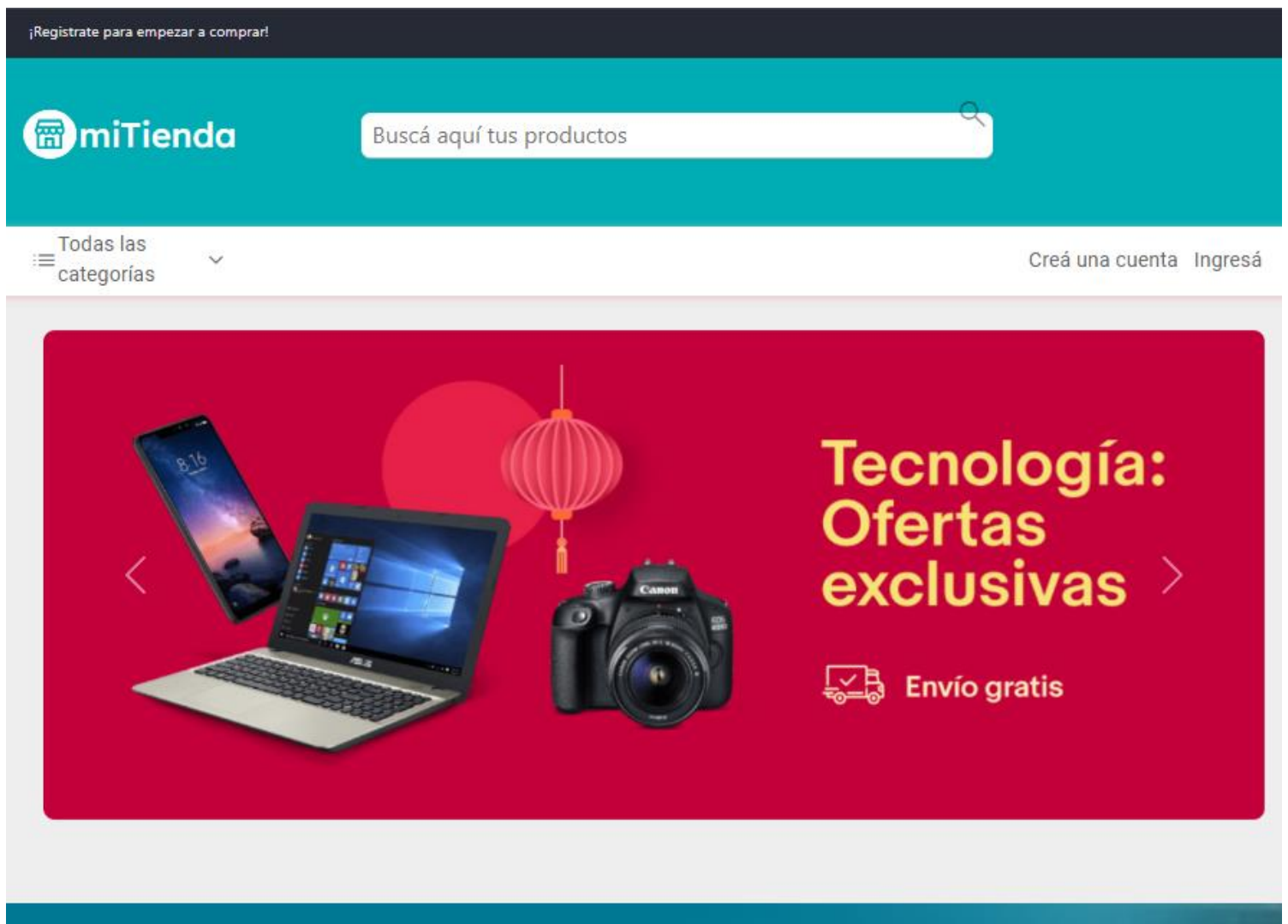
Este comando creará un usuario admin y varios usuarios clientes, así como también productos con la herramienta Faker

admin: [admin@ejemplo.com](mailto:admin@ejemplo.com) , password: 1234

11. Iniciar server

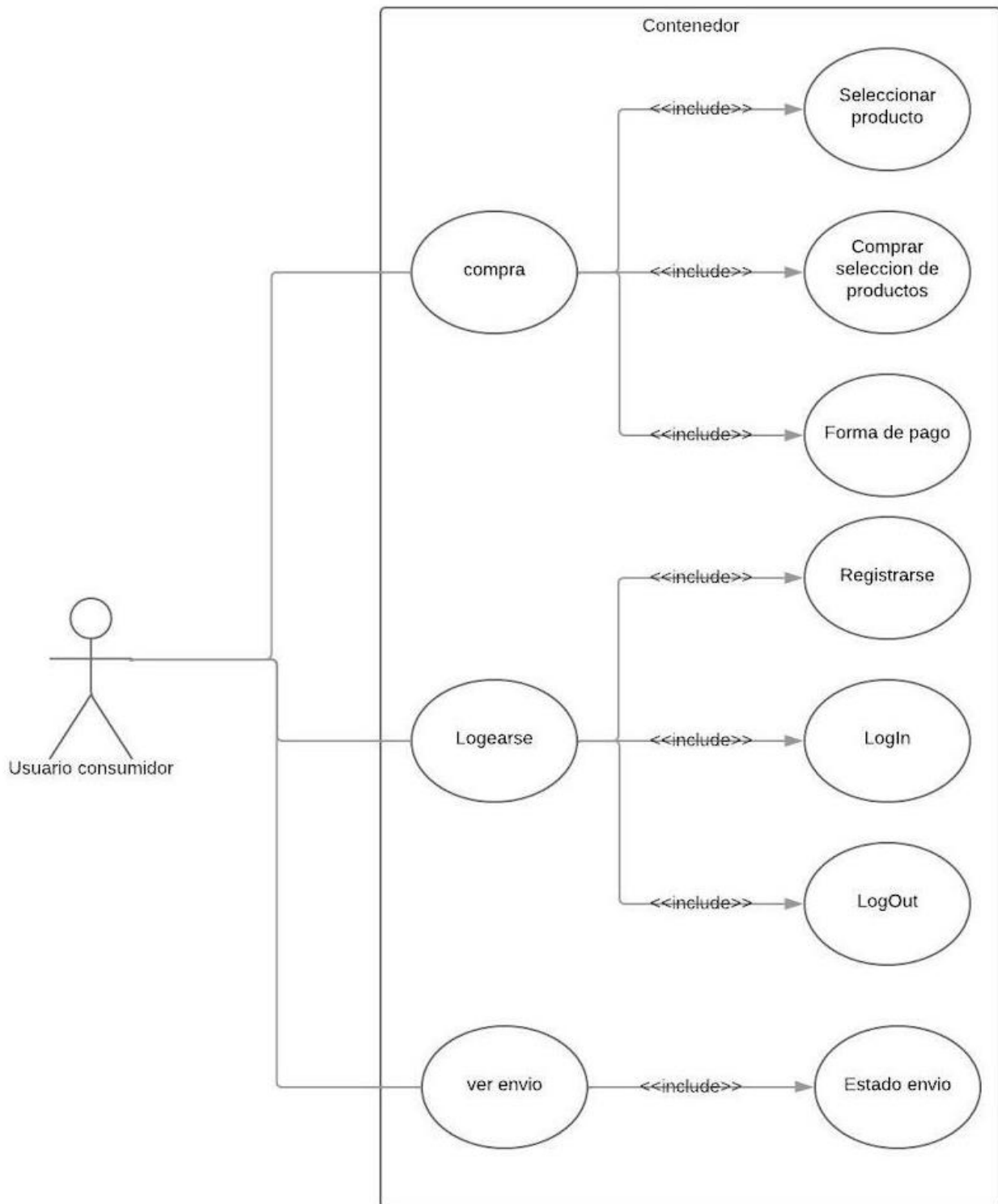
`php artisan serve`

12. Ir a **localhost:8000** en el navegador.



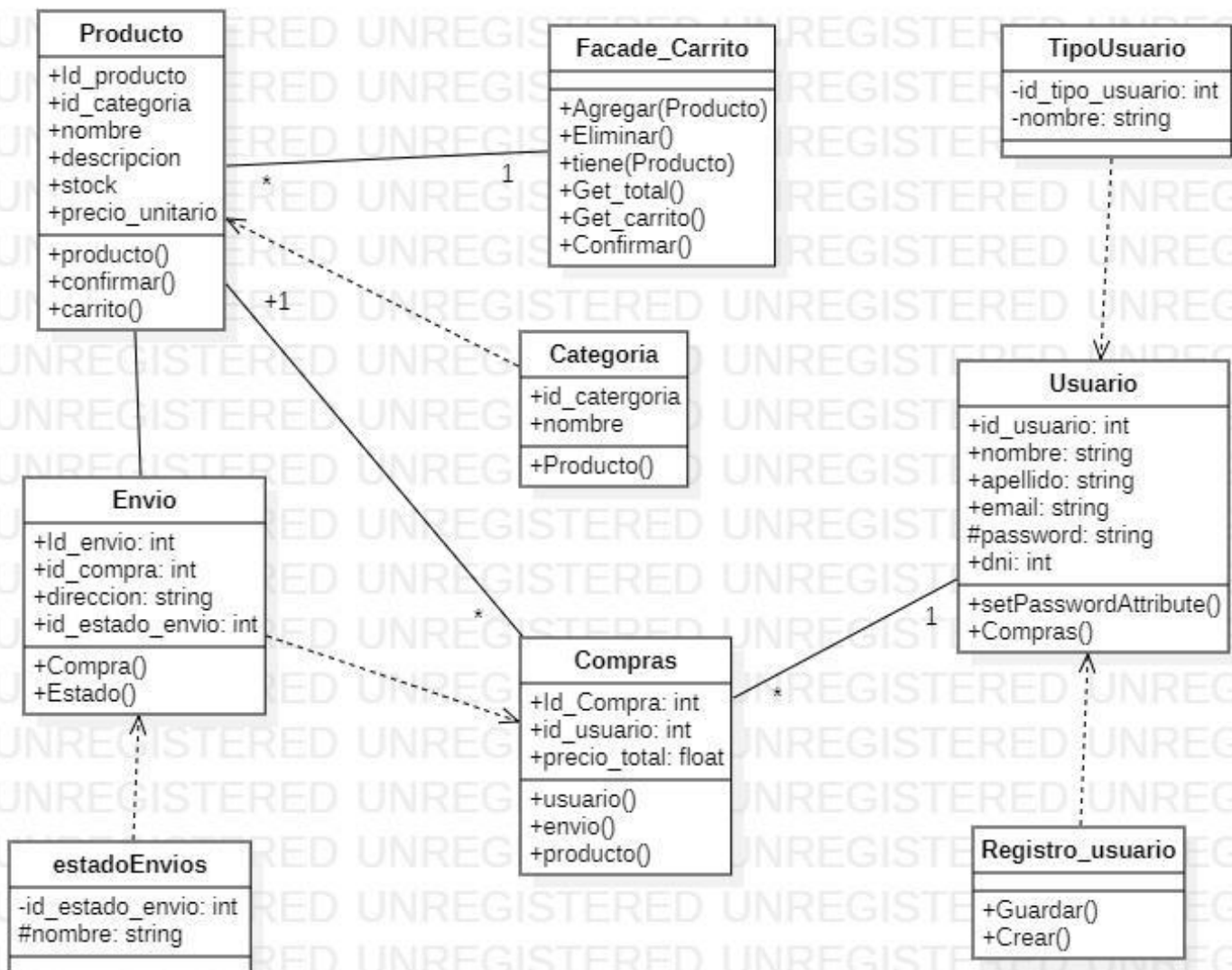
### 3. Diagramas del Software

#### 1. Diagrama de casos de uso

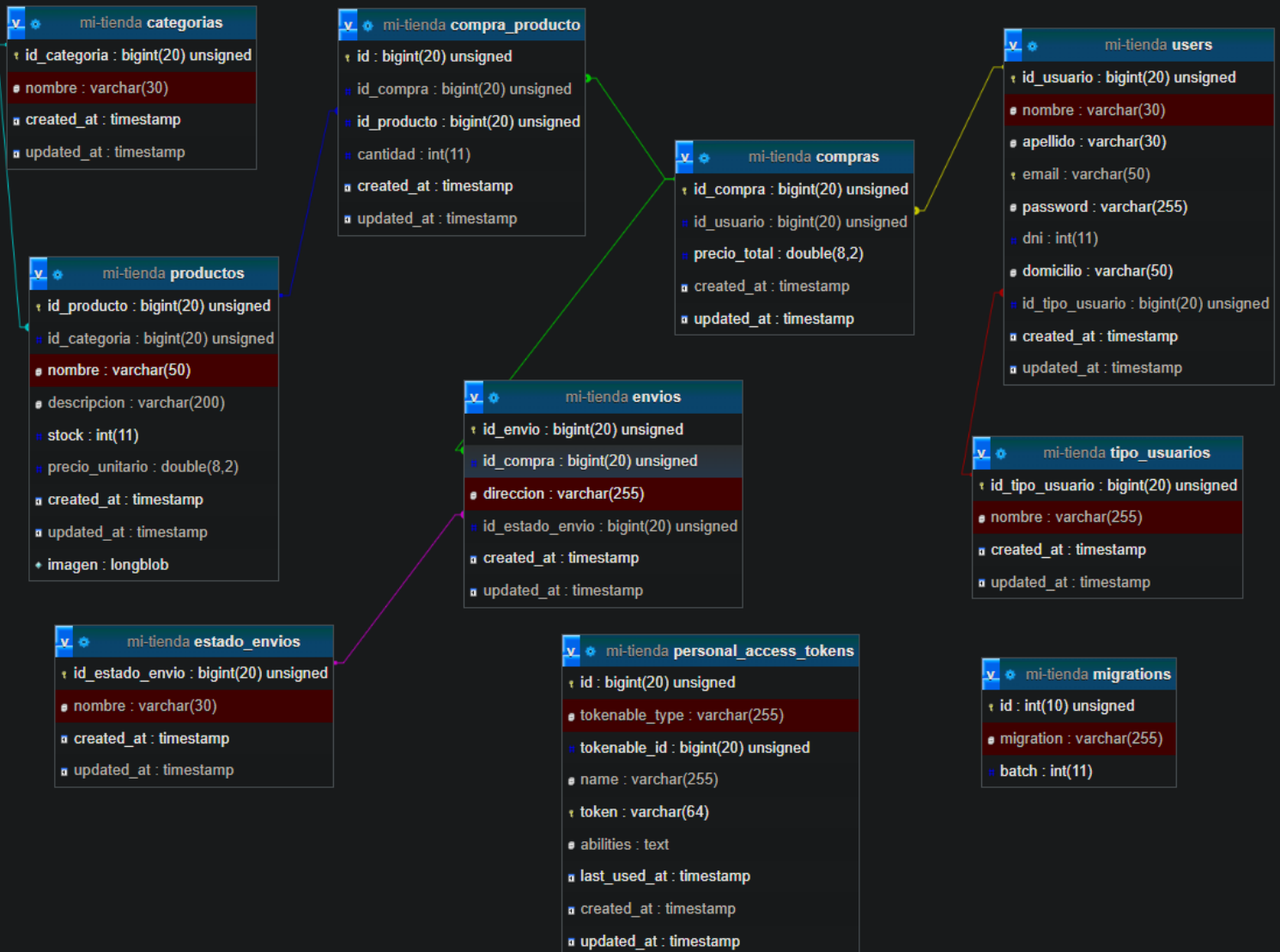




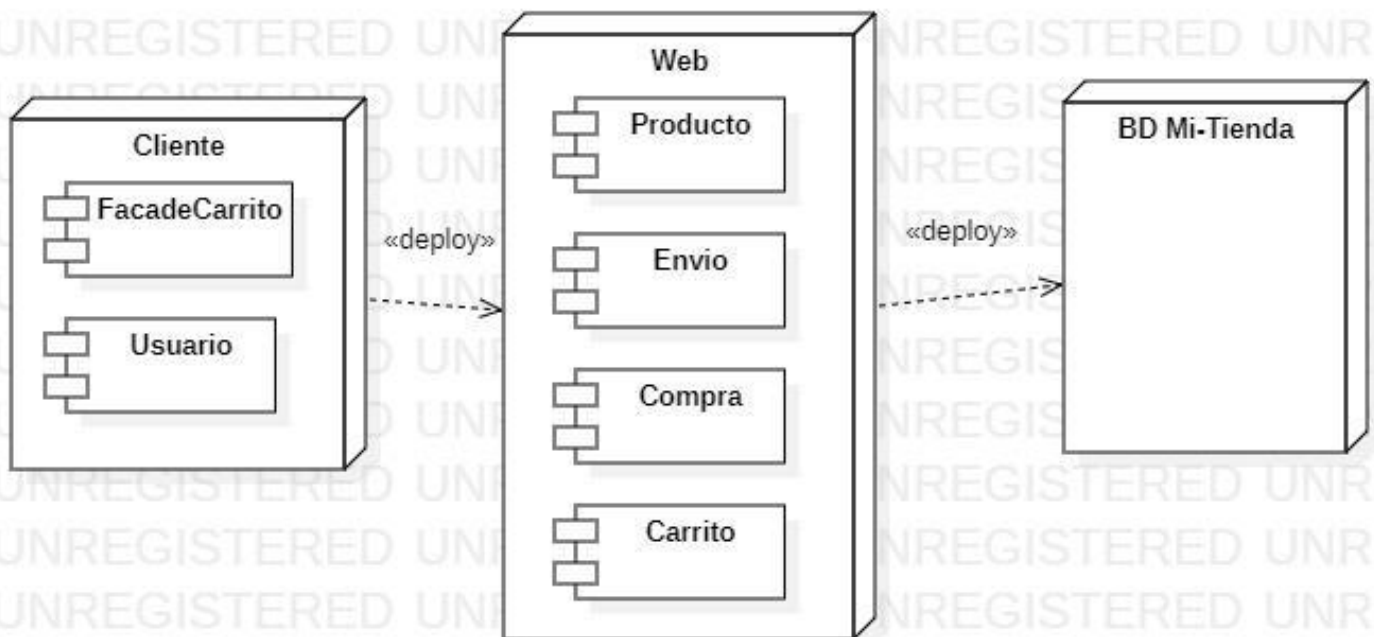
## 2. Diagrama de clases



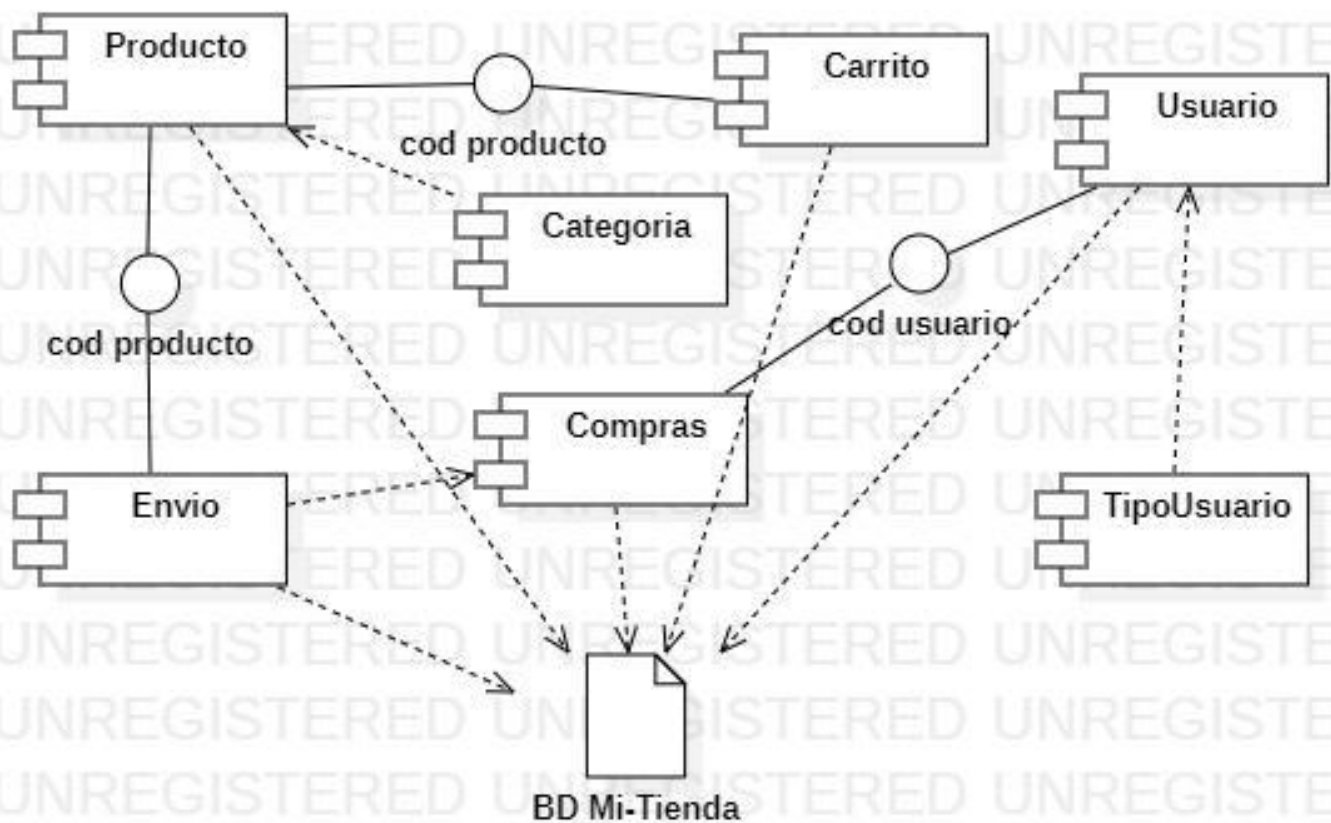
### 3. Diagrama entidad Relación



#### 4. Diagrama de componentes



#### 5. Diagrama de Despliegue



## 4. Propósito de la evaluación

El propósito es mejorar la calidad del producto mediante la aplicación de normas y estándares de calidad, como la norma ISO 9126 identificando y midiendo los diferentes atributos que son importantes para un software a nivel nacional e internacional.

## 5. Modelos de calidad

En la presente evaluación se evaluarán los modelos de calidad interna externa y de uso.

Modelos de calidad interna y externa son utilizados para evaluar la calidad del software en diferentes niveles.

Las métricas de calidad de uso están enfocadas en medir cómo los usuarios interactúan con el software y si el software cumple con sus necesidades.

### 5.1 Modelo de calidad interna:

El modelo de calidad interna se enfoca en la calidad del código fuente del software. Algunas métricas que se pueden utilizar para evaluar la calidad interna son:

**Complejidad ciclomática:** mide la complejidad del código y su capacidad de ser comprendido y mantenido.

**Cobertura de código:** mide la cantidad de código que es ejecutado durante las pruebas.

**Longitud de los métodos:** mide la cantidad de líneas de código que hay en cada método.

### 5.2 Modelo de calidad externa:

El modelo de calidad externa se enfoca en la calidad del software en términos de su funcionalidad y usabilidad. Algunas métricas que se pueden utilizar para evaluar la calidad externa son:

**Eficiencia:** mide la capacidad del software para realizar tareas en un tiempo adecuado.

**Fiabilidad:** mide la capacidad del software para funcionar correctamente y sin errores.

**Usabilidad:** mide la capacidad del software para ser utilizado fácilmente por los usuarios.

### 5.3 Métricas de calidad de uso:

Las métricas de calidad de uso se enfocan en evaluar la facilidad de uso y la satisfacción de los usuarios. Algunas métricas que se pueden utilizar para evaluar la calidad de uso son:

**Tiempo de aprendizaje:** mide el tiempo que le toma a un usuario aprender cómo utilizar el software.

**Eficiencia de uso:** mide la cantidad de tiempo que le toma a un usuario realizar una tarea utilizando el software.

**Satisfacción del usuario:** mide el grado de satisfacción que tiene el usuario al utilizar el software.

## 6. Selección de atributos para la calidad Interna

**6.1 Funcionalidad:** Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas.

**6.1.1 Precisión:** Capacidad del software para proporcionar efectos o resultados correctos o convenientes con el grado de exactitud necesario

**6.1.2 Seguridad (informática):** Capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.

**6.1.3 Exactitud:** Capacidad del software para proveer los resultados correctos y los efectos pactados, con un adecuado grado de precisión.

**6.1.4 Interoperabilidad:** Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.

**6.2 Confiabilidad:** La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas

**6.2.1 Madurez:** Capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.

- 6.2.2 Tolerancia a fallos:** Capacidad del producto de software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de la infracción de su interface especificada.
- 6.2.3 Recuperabilidad:** Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.
- 6.3 Eficiencia:** capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados usados, bajo condiciones establecidas
  - 6.3.1 Rendimiento:** capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas.
  - 6.3.2 Utilización de recursos:** capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas.
  - 6.3.3 Conformidad de la eficiencia:** Capacidad del producto de software de adherirse a las normas o convenciones que se relacionan con la eficiencia.

## 7 Selección de atributos para la calidad externa:

- 7.1 Usabilidad:** La capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas.
  - 7.1.1 Comprensibilidad:** La capacidad del producto para permitirle al usuario entender si el software es idóneo y como puede usarse para las tareas y condiciones de uso particulares.
  - 7.1.2 Cognoscibilidad:** La capacidad del producto para permitirle al usuario aprender su aplicación.
  - 7.1.3 Operabilidad:** La capacidad del producto para permitirle al usuario operarlo y controlarlo.
- 7.2 Mantenibilidad:** Capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.
  - 7.2.1 Diagnosticabilidad:** Capacidad del producto del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o para identificar las partes que van a ser modificadas.
  - 7.2.2 Flexibilidad:** Capacidad del producto del software para permitir la aplicación de una modificación especificada.
  - 7.2.3 Estabilidad:** Capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas al software.



**7.3 Portabilidad:** Capacidad de producto de software de ser transferido de un ambiente a otro.

**7.3.1 Adaptabilidad:** Capacidad del producto de software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines.

**7.3.2 Instalabilidad:** Capacidad del producto de software de ser instalado en un ambiente especificado.

**7.3.3 Coexistencia:** Capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.

## **8 Selección de atributos para la calidad de uso:**

**8.1 Calidad durante el uso:** capacidad del producto de software de permitir que los usuarios especificados alcancen los objetivos especificados con efectividad, productividad, seguridad y satisfacción en contextos de uso especificados.

**8.1.1 Eficacia:** capacidad del producto de software de permitir que los usuarios logren objetivos especificados con precisión e integridad en un contexto especificado.

**8.1.2 Productividad:** capacidad del producto de software de permitir que los usuarios dediquen una cantidad de recursos apropiada en relación con la eficacia alcanzada en un contexto de uso especificado.

**8.1.3 Seguridad:** capacidad del producto de software de alcanzar niveles aceptables de riesgo de daños a las personas, el negocio, el software, la propiedad o el ambiente en un contexto de uso especificado.

## **9 Métricas a utilizar Para la calidad interna:**

**9.1 Métricas de Funcionalidad:** Las métricas de funcionalidad se utilizan para predecir si el producto de software en cuestión satisface los requisitos funcionales prescritos y necesidades de los usuarios implicados.

**9.1.1 Métricas de Precisión:** indican un conjunto de atributos para evaluar la capacidad del producto de software para lograr el resultado correcto o agradable.

**9.1.2 Métricas de Seguridad:** Métricas de seguridad internas indican un conjunto de atributos para evaluar la capacidad del producto de software para evitar el acceso ilegal al sistema y / o datos.

**9.2 Métricas de Confiabilidad:**

**9.2.1 Métricas de Madurez:** Métricas de madurez interna indican un conjunto de atributos para evaluar la madurez de software.

**9.2.2 Métricas de tolerancia a fallos:** Métricas de tolerancia a fallos internos indican un conjunto de atributos para evaluar la capacidad de los productos de software en el mantenimiento de un nivel de rendimiento deseado en caso de fallos operativos o de la violación de su interfaz especificada.

### 9.3 Métricas de Eficacia:

**9.3.1 Métricas de rendimiento en el tiempo:** Métricas de comportamiento de tiempo internos indican un conjunto de atributos para predecir el comportamiento de tiempo del sistema de ordenador que incluye el producto de software durante la prueba o en funcionamiento.

**9.3.2 Métricas de utilización de recursos:** Métricas de utilización de recursos internos indican un conjunto de atributos para predecir la utilización de los recursos de hardware del sistema informático que incluye el producto de software durante las pruebas o en funcionamiento

## 10 . Métricas a utilizar Para la calidad externa:

**10.1.1 Usabilidad:** Métricas de usabilidad miden el grado en el que el software puede ser comprendido, aprendido, operado, atractivo y cumplen con las normas y directrices de usabilidad.

1. **Métricas de Comprensibilidad:** Los usuarios deben ser capaces de seleccionar un producto de software, que se adecuado para el uso previsto. Una estandaribilidad métrica un externo debe ser capaz de evaluar si los nuevos usuarios puedan entender si el software es adecuado, la forma en que se pueda utilizar para tareas particulares.
2. **Cognoscibilidad:** Métrica externa debe ser capaz de evaluar cuánto tiempo los usuarios tarda en aprender cómo utilizar las funciones particulares, y la eficacia de los sistemas de ayuda y documentación. Facilidad de aprendizaje está fuertemente relacionada con la comprensión, y mediciones comprensibilidad puede ser indicadores de la potencial capacidad de aprendizaje del software.

**10.1.2 Mantenibilidad:** Una métrica de mantenimiento externo debe ser capaz de medir atributos tales como el comportamiento del mantenedor, usuario o del sistema, incluyendo el software, cuando el software se mantiene o se modifica durante las pruebas o mantenimiento



1. **Diagnosticabilidad:** Una métrica externa de diagnosticabilidad debe ser capaz de medir estos atributos como el esfuerzo del usuario o gastado de los recursos
2. **Métricas de Flexibilidad:** Una métrica mutabilidad externo debe ser capaz de medir atributos tales como el esfuerzo al usuario el mantenedor o para medir el comportamiento del mantenedor, el usuario o sistema que incluye el software cuando intento ING para implementar una modificación especificado.
3. **Métricas de Estabilidad:** Una métrica estabilidad externa debe ser capaz de medir los atributos relacionados con un comportamiento inesperado del sistema, incluyendo el software cuando el software se prueba o operado después de la modificación.

#### 10.1.3 Portabilidad:

1. **Métricas de Adaptabilidad:** Una adaptabilidad externa métrica debe ser capaz de medir atributos tales como el comportamiento del sistema o el usuario que está tratando de adaptar el software a diferentes entornos especificados. Cuando un usuario tiene que aplicar un procedimiento de adaptación que no sea previamente proporcionada por el software para una necesidad de adaptación específica, debe ser medido el esfuerzo del usuario que requiera la adaptación.
2. **Métricas de Instalabilidad:** Una métrica instalabilidad externo debe ser capaz de medir atributos tales como el comportamiento del sistema o el usuario que está intentando instalar el software en un entorno específico del usuario.
3. **Métricas de coexistencia:** Una métrica coexistencia externo debe ser capaz de medir atributos tales como el comportamiento del sistema o el usuario que está intentando utilizar el software con otro software independiente en un entorno común de intercambio de recursos comunes.

## 11 . Métricas a utilizar para la calidad de Uso:

**11.1 Métricas calidad de uso:** Algunas métricas de usabilidad externos (ISO / IEC 9126-2) se prueban en una manera similar, pero evaluar el uso de características determinadas del producto durante el uso más general del producto para lograr una tarea típica como parte de una prueba de la calidad en el uso.

**11.2 Métricas de eficacia:** Métricas de efectividad evaluar si las tareas realizadas por los usuarios a lograr los objetivos especificados con exactitud e integridad en un contexto de uso especificado. No toman en cuenta cómo se lograron, sólo el grado en que se lograron los objetivos.

**11.2.1 Métricas de Productividad:** Métricas de productividad Evaluar los recursos que los usuarios consumen en relación con la eficacia

alcanzado en un contexto de uso especificado. El recurso más común es el tiempo para completar la tarea, aunque otros recursos pertinentes podrían incluir el esfuerzo del usuario, los materiales o el costo financiero de uso.

**11.2.2 Métricas de Seguridad:** Métricas de seguridad evaluar el nivel de riesgo de daño a las personas, las empresas, el software, los bienes o el medio ambiente en un contexto de uso especificado. Incluye la salud y seguridad del tanto al usuario como a los afectados por el uso, así como las consecuencias físicas o económicas imprevistas.

## 12. Tablas Métricas

### Métricas Funcionalidad externas

Características	subcaracterísticas	métricas	Propósito de la métrica	Formula	Interpretación del valor medido
Funcionalidad	Precisión	<b>La exactitud de las expectativas</b>	¿Son las diferencias entre los resultados esperados reales y razonables aceptables?	$X = A / T$ A =Número de casos encontrados por los usuarios con una diferencia frente a los resultados esperados razonables más allá permisible Tiempo T =Operación	$0 \leq X$ El más cercano a 0 es el mejor.
		<b>Exactitud Computacional</b>	¿Con qué frecuencia los usuarios finales se encuentran con resultados inexactos?	$X = A / T$ A =Número de casos encontrados por los usuarios con una diferencia frente a los resultados esperados razonables más allá permisible Tiempo T =Operación	$0 \leq X$ El más cercano a 0 es el mejor.
	Interoperabilidad	<b>Intercambiabilidad de datos (Formato de datos basada)</b>	Cómo tener correctamente las funciones de interfaz de intercambio para la transferencia de datos especificado puesto en práctica?	$X = A / B$ A =Número de formatos de datos que son aprobados para ser intercambiado con éxito con otro software o sistema durante las pruebas en el intercambio de datos, B = Número total de formatos de intercambio de datos	$0 \leq X \leq 1$ . Cuanto más se acerca a 1, el más correcto.

Características	subcaracterísticas	métricas	Propósito de la métrica	Formula	Interpretación del valor medido
		<b>Intercambiabilidad de datos (Intentó éxito del usuario basado)</b>	¿Con qué frecuencia el usuario final no logra intercambiar datos entre el software de destino y otro software?	a) $X = 1 - A / B$ A = Número de casos en que el usuario fracasan para intercambiar datos con otros programas o sistemas B = Número de casos en los que intento usuario para el intercambio de datos	$0 \leq X \leq 1$ . Cuanto más se acerca a 1, el más correcto.
	Seguridad	<b>Auditabilidad Acceso</b>	¿Qué tan completa es la pista de auditoría en relación con el acceso de los usuarios al sistema y sus datos?	$X = A / B$  A = Número de “usuario accede al sistema y los datos”,, grabado en la base de datos de historial de acceso B = Número de “usuario accede al sistema y los datos” realizado durante la evaluación	$0 \leq X \leq 1$ El más cercano a 1,0 es la mejor
		<b>Controlabilidad Acceso</b>	¿Cómo controlable es el acceso al sistema?	$X = A / B$  A = Número de detectados los diferentes tipos de operaciones ilegales B = Número de tipos de operaciones ilegales como en la especificación	$0 \leq X \leq 1$ El más cercano a 1,0 es la mejor

Características	subcaracterísticas	métricas	Propósito de la métrica	Formula	Interpretación del valor medido
Usabilidad	Facilidad de aprendizaje	Facilidad de aprendizaje función	¿Cuánto tarda el usuario tarda en aprender a utilizar una función?	T= tiempo medio que tarda	$0 < T$ Cuanto más corto es el mejor.
		Facilidad de aprendizaje para realizar una tarea en uso	¿Cuánto tarda el usuario en aprender a realizar la tarea especificada de manera eficiente?	T = Suma de tiempo de operación del usuario hasta que el usuario logra llevar acabo la tarea especificada en poco tiempo	$0 < T$ Cuanto más corto es el mejor.
	Comprensibilidad	Integridad de la descripción	¿Qué proporción de funciones (o tipos de funciones) se entiende después de leer la descripción del producto?	$X = A / B$ A = Número de demostraciones / tutoriales que el usuario acceda al éxito B = Número de demostraciones / tutoriales disponibles	$0 \leq X \leq 1$ El más cercano a 1,0 es la mejor
		Demostración una accesibilidad	¿Qué proporción de las demostraciones /tutoriales puede el usuario de acceso cada vez que el usuario realmente tiene que hacer durante la operación?	$X = A / B$ A = Número de casos en que los usuarios ver con éxitos demostración cuando el usuario intenta ver demostración B = Número de casos en que el usuario intenta ver demostración	$0 \leq X \leq 1$ El más cercano a 1,0 es la mejor

Características	subcaracterísticas	métricas	Propósito de la métrica	Formula	Interpretación del valor medido
				durante el período de observación	
	Atractivo	Interacción atractiva	¿Cómo de atractiva es la interfaz para el usuario?	Cuestionario para los usuarios	Dependerá de su método de puntaje cuestionario.
		Customizability apariencia Interface	¿Qué proporción de elementos de la interfaz se puede personalizar en apariencia a la satisfacción del usuario?	$X = A / B$  A =Número de elementos de la interfaz personalizada en apariencia a la satisfacción del usuario B = Número de elementos de la interfaz que el usuario desea personalizar	$0 \leq X \leq 1$ El más cercano a 1,0 es la mejor

## Métricas Funcionalidad internas

### 4.1. Métrica interna

- **Funcionalidad.** - Es la capacidad del software para poder satisfacer las funciones a las necesidades declaradas.
  - Precisión. – El software debe tener los resultados correctos con la exactitud necesaria.
  - Seguridad(Informática). – El software debe proteger la información y los datos, para que las personas no autorizadas no puedan modificar o leer los mismos
  - Cumplimiento. – El software debe un cumplimiento de calidad para dar un mejor funcionamiento
- **Confiabilidad.** - La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas
  - Madurez. – Es la capacidad del software para evitar un fallo total como resultado de haberse producido un fallo del software.
  - Tolerancia a fallos. – Capacidad del software de mantener un nivel de ejecución especificado en caso de fallos o infracción en su interfaz.
  - Recuperabilidad. - Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.

Características	Sub características	métrica	Propósito de la métrica	Fórmula	Interpretación del valor medido
Funcionalidad	Precisión	Precisión de datos	¿Con qué frecuencia se encuentran resultados con precisión inexacta?	$X = A / T$ A = Número de datos inexactos encontrados Tiempo T = Operación	$0 \leq X$ El más Cercano a 0 es el más óptimo.
		Exactitud del software	¿Con qué frecuencia se encuentran inexactitudes de datos en el software?	$X = A / T$ A = Número de datos inexactos encontrados Tiempo T = Operación	$0 \leq X$ El más Cercano a 0 es el más óptimo.
	Seguridad	Auditabilidad Acceso	Cuenta el número de tipos de acceso que se está registrando correctamente como en las especificaciones y comparar con el número de tipos de acceso que son necesario para estar conectado con las especificaciones?	$X = A / B$ A = Número de tipos de acceso que se está registrando como en las especificaciones B = Número de tipos de acceso necesarias para estar conectado con las especificaciones	$0 \leq X \leq 1$ Cuanto más se acerca a 1, la más auditable.



Características	Sub características	métrica	Propósito de la métrica	Fórmula	Interpretación del valor medido
	Cumplimiento	Cumplimiento Funcional	¿Cómo es compatible con la funcionalidad del producto a los reglamentos, normas y convenciones.	$X = A / B$ A = Número de artículos correctos relacionados con el cumplimiento de la funcionalidad B = número total de artículos de cumplimiento	$0 \leq X \leq 1$ . Cuanto más se acerca a 1, el más obediente.
		El cumplimiento estándar del sistema de la Interfaz	¿Cómo cumple son las interfaces con los reglamentos, normas y convenciones?	$X = A / B$ A = Número de interfaces implementadas correctamente como se especifica, confirmó en la revisión B = Número total de interfaces que requieren el cumplimiento	$0 \leq X \leq 1$ . Cuanto más se acerca a 1, el más obediente.
Fiabilidad	Madurez	Detección de fallos	¿Cuántos fallos fueron detectados?	$X = A / B$ A = absoluto número de fallas detectadas en la revisión B = Número de fallas estimados para ser detectadas en la revisión	$0 \leq X$ Un valor alto para X implica buena calidad del producto, mientras que A=0 no implica

Características	Sub características	métrica	Propósito de la métrica	Fórmula	Interpretación del valor medido
					necesariamente el estado libre de avería del artículo revisado.
		Densidad de fracaso frente a los casos de prueba	¿Cuántos fracasos fueron detectados durante el periodo de pruebas ?	$X = A1 / A2$ A1 = número de errores detectados A2 = número de casos de prueba realizados	$0 \leq X$ Depende de las pruebas. Las pruebas mientras más pequeño mejor.
	Tolerancia a Fallos	Evitación de fracasos	¿Cuántos patrones de falla fueron traídos bajo control para evitar fallos críticos y serios?	$X = A / B$ A = Número de evitar sucesos críticos y serios fallo contra los casos de prueba de patrón de culpa B = Número de casos de prueba ejecutados de patrón de culpa (casi provocando fallos) durante la prueba	$0 \leq X \leq 1$ El más cercano a 1,0 es mejor, ya que el usuario más A menudo Puede evitar el fracaso crítico o grave.
		Evitar la operación n incorrecta	¿Cuántas funciones implementadas con las	$X = A / B$ A = Número de fallos críticos y graves evitadas	$0 \leq X \leq 1$ El más cercano a 1,0 es el mejor, ya que se evita la

Características	Sub características	métrica	Propósito de la métrica	Fórmula	Interpretación del valor medido
			operaciones tienen la capacidad de evitar errores?	ocurrencias B = Número de casos de prueba ejecutados de patrones de funcionamiento incorrectos (casi provocando fallos) durante la prueba	operación de usuario más incorrecta.
	Recuperabilidad	Restaurabilidad	¿Qué tan capaz es el producto en sí mismo después de la restauración de evento anormal o en la petición?	$X = A / B$ A = Número de requisitos de restauración aplicados confirmados en la revisión B = Número de requisitos de restauración en las especificaciones	$0 \leq X \leq 1$ Donde X es mayor, mejor restaurabilidad
		Efectividad Restauración	¿Qué tan efectiva es la capacidad de la restauración?	$X = A / B$ A = Número de requisitos de restauración aplicados reuniones objetivo tiempo de restauración B = Número de requisitos de restauración con tiempos objetivo especificados	$0 \leq X \leq 1$ Donde X es mayor, mejor eficacia

## Métricas de uso

Características	métricas	Propósito de la métrica	Formula	Público objetivo
Eficacia	<b>La eficacia de tareas</b>	¿Qué proporción de los objetivos de la tarea se logra correctamente?	$M1 =  1 - \odot A $	Usuario Diseñador de interfaz humana
	<b>Realización de tareas</b>	¿Qué proporción de las tareas se han completado?	$X = A / B$	Usuario Diseñador de interfaz humana
	<b>Frecuencia de error</b>	¿Cuál es la frecuencia de los errores?	$X = A / T$	Usuario Diseñador de interfaz humana
productividad	<b>Tiempo de tareas</b>	¿Cuánto tiempo se tarda en completar una tarea?	$X = A / B$	Usuario Diseñador de interfaz humana
	<b>La eficiencia de tareas</b>	¿Qué tan eficiente son los usuarios?	$X = M1 / T$	Usuario
	<b>La productividad económica</b>	¿Qué tan efectivo costo es el usuario?	$X = M1 / C$	Usuario
	<b>Proporción Productivo</b>	¿Qué proporción de las veces es el usuario que realiza acciones productivas?	$X = A / B$	Usuario
	<b>La eficiencia relativa de usuario</b>	¿Qué tan eficiente es un usuario en comparación con un experto?	$X = Ta / Tb$	Usuario
seguridad	<b>La salud y la seguridad del usuario</b>	¿Cuál es la incidencia de problemas de salud entre los usuarios del producto?	$X = 1 - A / B$	Usuario

Características	métricas	Propósito de la métrica	Formula	Público objetivo
	<b><i>Seguridad de las personas afectadas por el uso del sistema</i></b>	¿Cuál es la incidencia de peligro para las personas afectadas por el uso del sistema?	$X = 1 - A / B$	Usuario
	<b><i>Daños económicos</i></b>	¿Cuál es la incidencia de daño económico?	$X = 1 - A / B$	Usuario
	<b><i>Daños Software</i></b>	¿Cuál es la incidencia de la corrupción de software?	$X = 1 - A / B$	Usuario
satisfacción	<b><i>Escala de satisfacción</i></b>	¿Qué tan satisfecho está el usuario?	$X = A / B$	Usuario
	<b><i>Cuestionario de satisfacción</i></b>	¿Qué tan satisfecho está el usuario con las características específicas de software?	$X = (A_i / n$	Usuario
	<b><i>El uso discrecional</i></b>	¿Qué proporción de los posibles usuarios optar por utilizar el sistema?	$X = A / B$	Usuario