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- )

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

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

SUCRE-BOLIVIA

Contenido

[1. Introducción 5](#_Toc130323010)

[2. Requisitos para la instalación 6](#_Toc130323011)

[3. Diagramas del Software 8](#_Toc130323012)

[Diagrama de casos de uso 8](#_Toc130323013)

[Diagrama de clases 9](#_Toc130323014)

[Diagrama entidad Relación 10](#_Toc130323015)

[Diagrama de componentes 11](#_Toc130323016)

[Diagrama de Despliegue 11](#_Toc130323017)

[4. Propósito de la evaluación 12](#_Toc130323018)

[5. Modelos de calidad 12](#_Toc130323019)

[5.1 Modelo de calidad interna: 12](#_Toc130323020)

[5.2 Modelo de calidad externa: 12](#_Toc130323021)

[5.3 Métricas de calidad de uso: 13](#_Toc130323022)

[6. Selección de atributos para la calidad Interna 13](#_Toc130323023)

[6.1 Funcionalidad: 13](#_Toc130323024)

[**6.1.1** Precisión: 13](#_Toc130323025)

[**6.1.2** Seguridad (informática): 13](#_Toc130323026)

[**6.1.3** Exactitud: 13](#_Toc130323027)

[**6.1.4** Interoperabilidad 13](#_Toc130323028)

[6.2 Confiabilidad: 13](#_Toc130323029)

[**6.2.1** Madurez: 13](#_Toc130323030)

[**6.2.2** Tolerancia a fallos: 14](#_Toc130323031)

[**6.2.3** Recuperabilidad: 14](#_Toc130323032)

[6.3 Eficiencia 14](#_Toc130323033)

[**6.3.1** Rendimiento: 14](#_Toc130323034)

[**6.3.2** Utilización de recursos: 14](#_Toc130323035)

[**6.3.3** Conformidad de la eficiencia: 14](#_Toc130323036)

[**7** Selección de atributos para la calidad externa 14](#_Toc130323037)

[7.1 Usabilidad 14](#_Toc130323038)

[**7.1.1** Comprensibilidad: 14](#_Toc130323039)

[**7.1.2** Cognoscibilidad: 14](#_Toc130323040)

[**7.1.3** Operabilidad: 14](#_Toc130323041)

[7.2 Mantenibilidad 14](#_Toc130323042)

[**7.2.1** Diagnosticabilidad 14](#_Toc130323043)

[**7.2.2** Flexibilidad 14](#_Toc130323044)

[**7.2.3** Estabilidad 14](#_Toc130323045)

[7.3 Portabilidad 14](#_Toc130323046)

[**7.3.1** Adaptabilidad 15](#_Toc130323047)

[**7.3.2** Instalabilidad 15](#_Toc130323048)

[**7.3.3** Coexistencia: 15](#_Toc130323049)

[**8** Selección de atributos para la calidad de uso 15](#_Toc130323050)

[8.1 Calidad durante el uso: 15](#_Toc130323051)

[**8.1.1** Eficacia 15](#_Toc130323052)

[**8.1.2** Productividad 15](#_Toc130323053)

[**8.1.3** Seguridad 15](#_Toc130323054)

[**9** Métricas a utilizar Para la calidad interna 15](#_Toc130323055)

[9.1 Métricas de Funcionalidad 15](#_Toc130323056)

[**9.1.1** Métricas de Precisión 15](#_Toc130323057)

[**9.1.2** Métricas de Seguridad 15](#_Toc130323058)

[9.2 Métricas de Confiabilidad 15](#_Toc130323059)

[**9.2.1** Métricas de Madurez 15](#_Toc130323060)

[**9.2.2** Métricas de tolerancia a fallos 15](#_Toc130323061)

[9.3 Métricas de Eficacia 16](#_Toc130323062)

[**9.3.1** Métricas de rendimiento en el tiempo 16](#_Toc130323063)

[**9.3.2** Métricas de utilización de recursos 16](#_Toc130323064)

[**10** Métricas a utilizar Para la calidad externa 16](#_Toc130323065)

[**10.1.1** Usabilidad 16](#_Toc130323066)

[**10.1.1.1** Métricas de Comprensibilidad 16](#_Toc130323067)

[**10.1.1.2** Cognoscibilidad 16](#_Toc130323068)

[**10.1.2** Mantenibilidad 16](#_Toc130323069)

[**10.1.2.1** Diagnosticabilidad 17](#_Toc130323070)

[**10.1.2.2** Métricas de Flexibilidad 17](#_Toc130323071)

[**10.1.2.3** Métricas de Estabilidad 17](#_Toc130323072)

[**10.1.3** Portabilidad 17](#_Toc130323073)

[**10.1.3.1** Métricas de Adaptabilidad 17](#_Toc130323074)

[**10.1.3.2** Métricas de Instalabilidad: 17](#_Toc130323075)

[**10.1.3.3** Métricas de coexistencia 17](#_Toc130323076)

[**11** Métricas a utilizar para la calidad de Uso 17](#_Toc130323077)

[11.1 Métricas calidad de uso 17](#_Toc130323078)

[**11.1.1** Métricas de eficacia 17](#_Toc130323079)

[**11.1.2** Métricas de Productividad 18](#_Toc130323080)

[**11.1.3** Métricas de Seguridad 18](#_Toc130323081)

Documentación del software para hacer la evaluación



# 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.

# Requisitos para la instalación

**Requerimientos**

* [**Laravel**](https://laravel.com/)
* [**NodeJs**](https://nodejs.org/es/)
* [**Composer**](https://getcomposer.org/)
* [**Php**](https://www.php.net/)
* [**Mysql**](https://www.mysql.com/)

Opcional para Php y Mysql

* [**XAMPP**](https://www.apachefriends.org/es/index.html)

**Instalación**

Para poder correr el repositorio localmente:

1. Clonar el repositorio

git clone https://github.com/El-Posho-Web/Mi-Tienda.git

1. Ir a la raiz del directorio

cd Mi-Tienda

1. Copiar el archivo .env.example a .env

cp .env.example .env

1. Crear la base de datos mi-tienda (puedes cambiar el nombre de la base de datos)
2. Go to .env file
   * Setear las credenciales de la base de datos (DB\_DATABASE=mi-tienda, DB\_USERNAME=root, DB\_PASSWORD=)
3. Instalar las dependencias de PHP

composer install

1. Generar la key

php artisan key:generate

1. Instalar las dependencias de Front-End

npm install && npm run dev

1. Correr las migraciones

php artisan migrate

1. Correr el seeder

php artisan db:seed

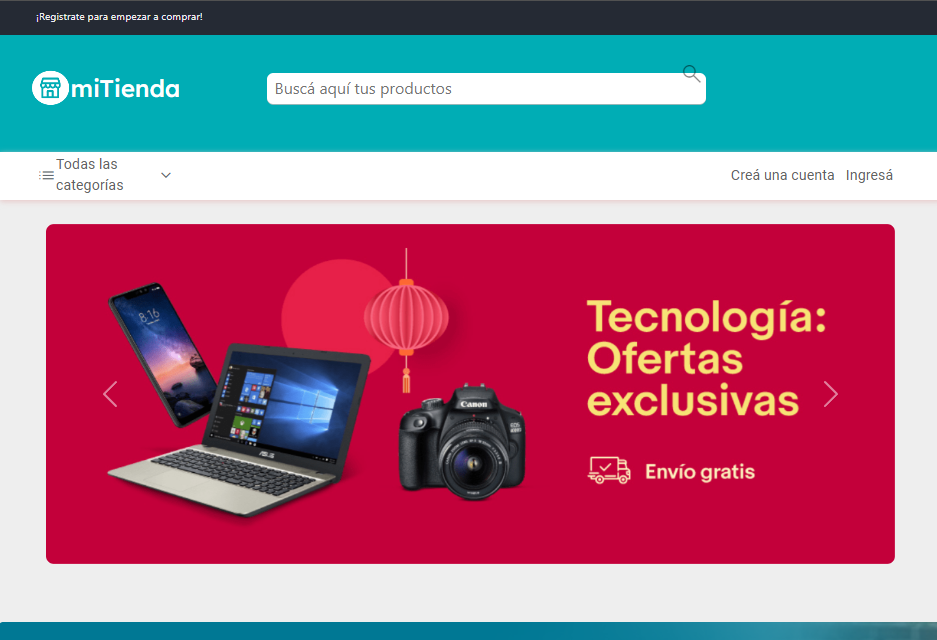
Este comando creara un usuario admin y varios usuarios clientes, asi como tambien productos con la herramienta Faker

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

1. Iniciar server

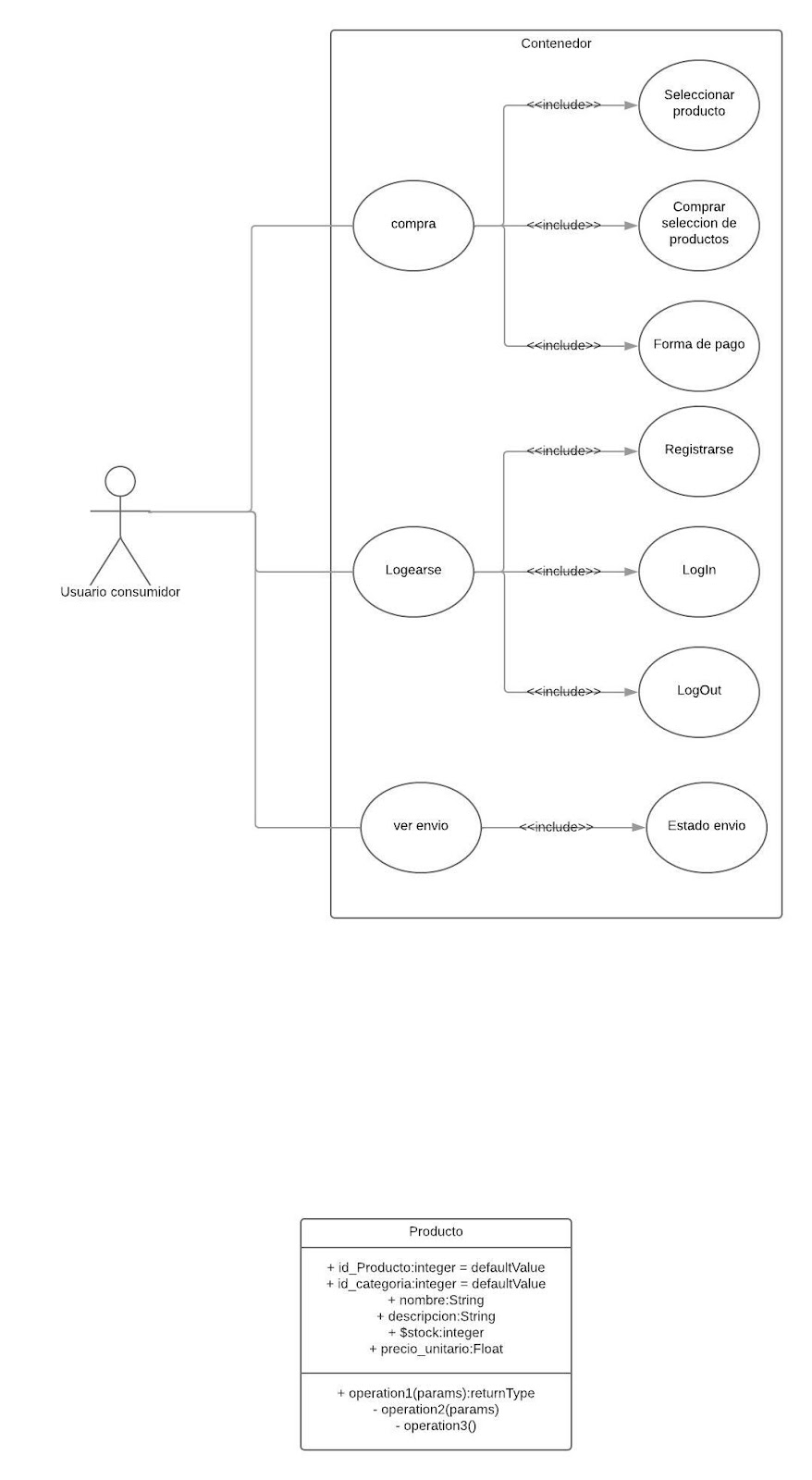
php artisan serve

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

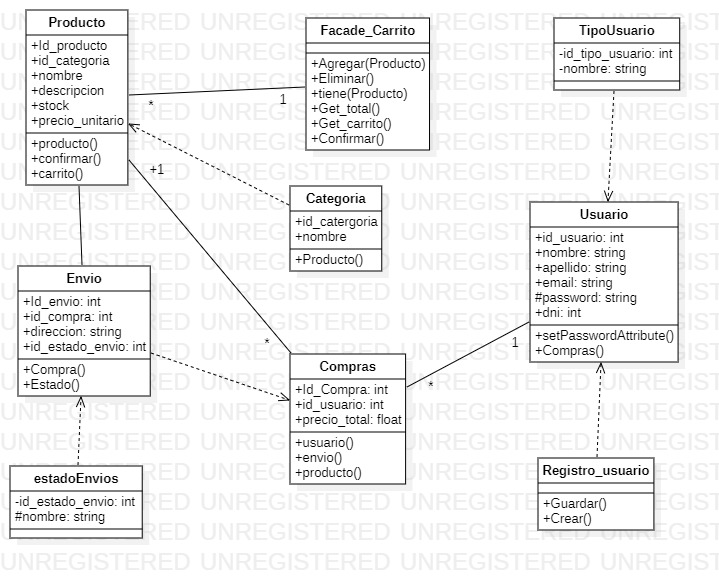


# Diagramas del Software

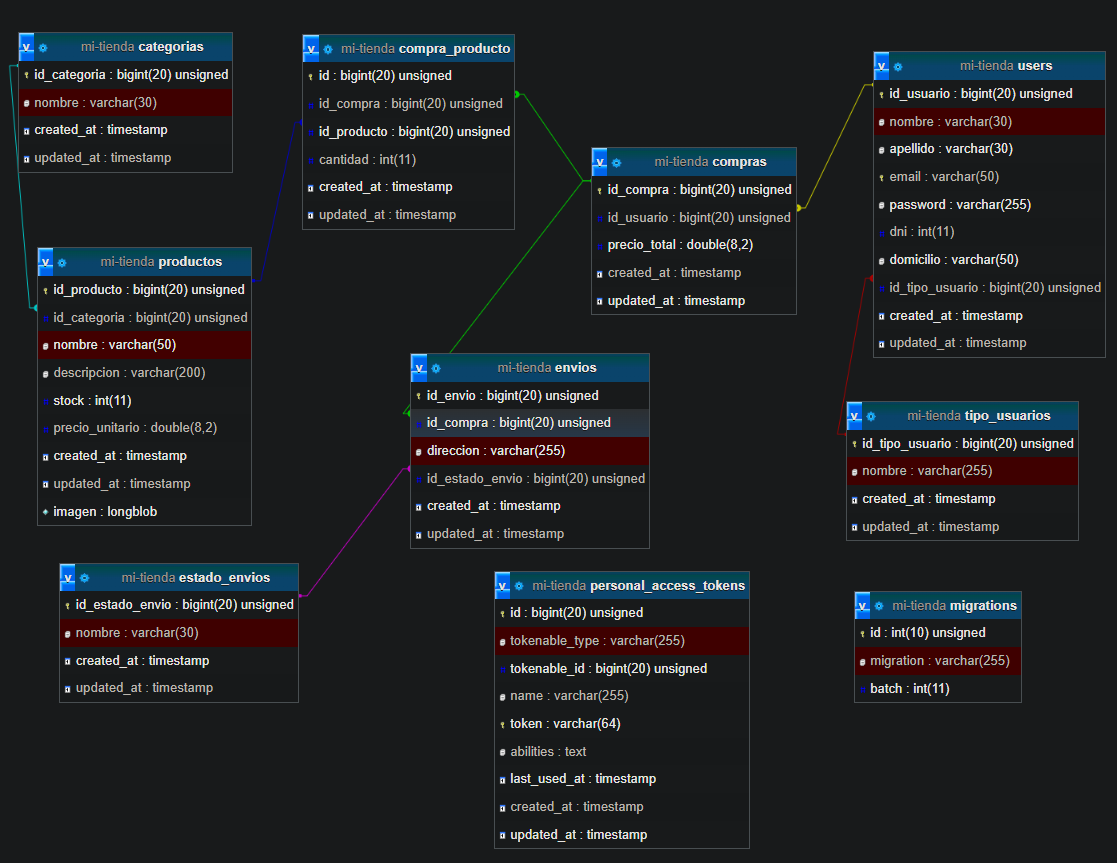
## Diagrama de casos de uso



## Diagrama de clases

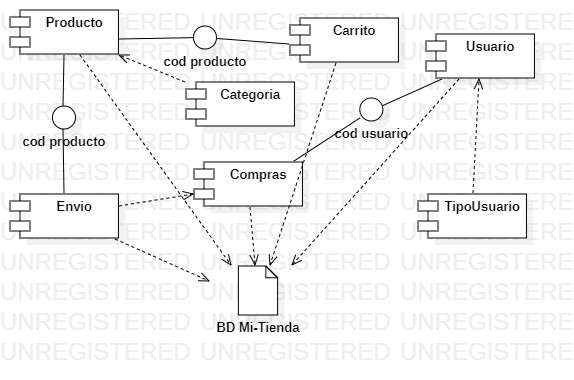


## Diagrama entidad Relación



## Diagrama de componentes

## Diagrama de Despliegue



# 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.

# 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.

## 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.

## 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.

## 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.

1. Selección de atributos para la calidad Interna

* 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.
     1. Precisión: Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario
     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.
     3. Exactitud:Capacidad del software para proveer los resultados correctos y los efectos pactados, con un adecuado grado de precisión.
     4. Interoperabilidad: Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.
  2. Confiabilidad:La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas
     1. Madurez:Capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.
     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.
     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.
  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
     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.
     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.
     3. Conformidad de la eficiencia: Capacidad del producto de software de adherirse a las normas o convenciones que se relacionan con la eficiencia.

1. Selección de atributos para la calidad externa**:**
   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.
      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.
      2. Cognoscibilidad: La capacidad del producto para permitirle al usuario aprender su aplicación.
      3. Operabilidad: La capacidad del producto para permitirle al usuario operarlo y controlarlo.
   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.
      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.
      2. Flexibilidad**:** Capacidad del producto del software para permitir la aplicación de una modificación especificada.
      3. Estabilidad**:** Capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas al software.
   3. Portabilidad**:** Capacidad de producto de software de ser transferido de un ambiente a otro.
      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.
      2. Instalabilidad**:** Capacidad del producto de software de ser instalado en un ambiente especificado.
      3. Coexistencia:Capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.
2. Selección de atributos para la calidad de uso**:**
   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.
      1. Eficacia**:** capacidad del producto de software de permitir que los usuarios logren objetivos especificados con precisión e integridad en un contexto especificado.
      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.
      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.
3. Métricas a utilizar Para la calidad interna**:**
   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.
      1. Métricas de Precisión**:** indican un conjunto de atributos para evaluar la capacidad del producto de software para lograr el resultado correcto o agradable.
      2. Métricas de Seguridad**:** Métricas de seguridad internas indican un conjunto de atributo s para evaluar la capacidad del producto de software para evitar el acceso ilegal al sistema y / o datos.
   2. Métricas de Confiabilidad**:**
      1. Métricas de Madurez**:** Métricas de madurez interna indican un conjunto de atributos para evaluar la madurez de software.
      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.
   3. Métricas de Eficacia**:**
      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.
      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
4. . Métricas a utilizar Para la calidad externa**:**
   * 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.
     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.
     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.
5. . Métricas a utilizar para la calidad de Uso**:**
   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.
   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.
      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.
      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.

Métricas Funcionalidad externas

# Tablas Métricas

Métricas Funcionalidad externas

| **Características** | **subcaracterísticas** | **métricas** | **Propósito de la métrica** | **Formula** | **Interpretación del valor medido** |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Precisión | **La exactitud de las expectativas** |  | X = A / T |  |
|  |  | **Exactitud Computacional** |  | X = A / T |  |
|  | Interoperabilidad | **Intercambiabilidad de datos  (Formato de datos basada)** |  | X = A / B |  |
|  |  | **Intercambiabilidad de datos  (Intento éxito del usuario basado)** |  | X = 1 - A / B |  |
|  | Seguridad | **Auditabilidad Acceso** |  | X = A / B |  |
|  |  | **Controlabilidad Acceso** |  | X = A / B |  |
| Usabilidad | Facilidad de aprendizaje | **Facilidad de aprendizaje función** |  | T= tiempo medio que tarda |  |
|  |  | **Facilidad de aprendizaje para realizar una tarea en uso** |  | T= tiempo medio que tarda |  |
|  | Comprensibilidad | **Integridad de la descripción** |  | X = A / B |  |
|  |  | **Demostración una ccesibilidad** |  | X = A / B. |  |
|  | Atractivo | **Interacción atractiva** |  | Cuestionario para los usuarios |  |
|  |  | **Customisability apariencia Interface** |  | X = A / B |  |

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 <= 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 <= X El más  Cercano a 0  es el más óptimo. |
| Seguridad | Auditabilidad  Acceso | Cuente 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 <= X <= 1  Cuanto más se  acerca a 1, la  más auditable. |
|  |  |  |  |
| 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 <= X<= 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 <= X<= 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 <= X  Un valor alto  para X implica  buena calidad  del producto,  mientras que A=0 no implica  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<=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<=X<=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 operaciones tienen la capacidad de evitar errores? | X = A / B  A = Número de fallos críticos y graves evitadas  ocurrencias  B = Número de casos de prueba ejecutados de  patrones de funcionamiento incorrectos (casi  provocando fallos) durante la prueba | 0<=X<=1  El más cercano a 1,0 es el mejor, ya que se evita la 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  aplicadas confirmados en la revisión  B=Número de requisitos de restauración en  las especificaciones | 0 <= X <= 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  aplicadas reuniones objetivo tiempo de  restauración  B = Número de requisitos de restauración con  tiempos objetivo especificados | 0 <= X <= 1  Donde X es  mayor, mejor  eficacia |

Métricas de uso

| **Características** | **métricas** | **Propósito de la métrica** | **Formula** | Público objetivo |
| --- | --- | --- | --- | --- |
| Eficacia | ***La eficacia de tareas*** | ¿Qué proporción de los objetivos de la tarea se logra correctamente? | M1 = | 1- Σ A i | | Usuario  Diseñador de interfaz humana |
|  | ***Realización de tareas*** | ¿Qué proporción de las tareas se han completado? | X = A / B | Usuario  Diseñador de interfaz humana |
|  | ***Frecuencia de error*** | ¿Cuál es la frecuencia de los errores? | X = A / T | Usuario  Diseñador de interfaz humana |
| productividad | ***Tiempo de tareas*** | ¿Cuánto tiempo se tarda en completar una tarea? | X = A / B | Usuario  Diseñador de interfaz humana |
|  | ***La eficiencia de tareas*** | ¿Qué tan eficiente son los usuarios? | X = M1 / T | Usuario |
|  | ***La productividad económica*** | ¿Qué tan efectivo costo es el usuario? | X = M1 / C | Usuario |
|  | ***Proporción Productivo*** | ¿Qué proporción de las veces es el usuario que realiza acciones productivas? | X = A / B | Usuario |
|  | ***La eficiencia relativa de usuario*** | ¿Qué tan eficiente es un usuario en comparación con un experto? | X = Ta / Tb | Usuario |
| seguridad | ***La salud y la seguridad del usuario*** | ¿Cuál es la incidencia de problemas de salud entre los usuarios del producto? | X = 1-A / B | Usuario |
|  | ***Seguridad de las personas afectadas por el uso del sistema*** | ¿Cuál es la incidencia de peligro para las personas afectadas por el uso del sistema? | X = 1-A / B | Usuario |
|  | ***Daños económicos*** | ¿Cuál es la incidencia de daño económico? | X = 1-A / B | Usuario |
|  | ***Daños Software*** | ¿Cuál es la incidencia de la corrupción de software? | X = 1-A / B | Usuario |
| satisfacción | ***Escala de satisfacción*** | ¿Qué tan satisfecho está el usuario? | X = A / B | Usuario |
|  | ***Cuestionario de satisfacción*** | ¿Qué tan satisfecho está el usuario con las características específicas de software? | X = ∑ (A i) / n | Usuario |
|  | ***El uso discrecional*** | ¿Qué proporción de los posibles usuarios optar por utilizar el sistema? | X = A / B | Usuario |