****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Aplicación para la Generación Automatizada de un Diccionario de Datos**

Curso: Patrones de Software

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

Daleska Nicolle Fernandez Villanueva            (2021070308)

Andree Sebastián Flores Meléndez                (2017057494)

Mario Antonio Flores Ramos                           (2018000597)

**Tacna – Perú**

**2025**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | AFM | ELV | ARV | 10/10/2020 | Versión Original |

Aplicación para la Generación Automatizada

de un Diccionario de Datos

Documento de Arquitectura de Software

Versión 1.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

Contenido

[1. INTRODUCCIÓN 3](#_Toc196562644)

[1.1. Propósito (Diagrama 4+1) 4](#_Toc196562645)

[1.2. Alcance 4](#_Toc196562646)

[1.3. Definición, siglas y abreviaturas 4](#_Toc196562647)

[2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS 4](#_Toc196562648)

[2.1.1. Requerimientos Funcionales 5](#_Toc196562649)

[2.1.2. Requerimientos No Funcionales – Atributos de Calidad 5](#_Toc196562650)

[3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 5](#_Toc196562651)

[3.1. Vista de Caso de uso 5](#_Toc196562652)

[3.1.1. Diagramas de Casos de uso 6](#_Toc196562653)

[3.2. Vista Lógica 6](#_Toc196562654)

[3.2.1. Diagrama de Subsistemas (paquetes) 6](#_Toc196562655)

[3.2.2. Diagrama de Secuencia (vista de diseño) 6](#_Toc196562656)

[3.2.3. Diagrama de Colaboración (vista de diseño) 6](#_Toc196562657)

[3.2.4. Diagrama de Objetos 6](#_Toc196562658)

[3.2.5. Diagrama de Clases 6](#_Toc196562659)

[3.2.6. Diagrama de Base de datos (relacional o no relacional) 7](#_Toc196562660)

[3.3. Vista de Implementación (vista de desarrollo) 7](#_Toc196562661)

[3.3.1. Diagrama de arquitectura software (paquetes) 7](#_Toc196562662)

[3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes) 7](#_Toc196562663)

[3.4. Vista de procesos 7](#_Toc196562664)

[3.4.1. Diagrama de Procesos del sistema (diagrama de actividad) 7](#_Toc196562665)

[3.5. Vista de Despliegue (vista física) 7](#_Toc196562666)

[3.5.1. Diagrama de despliegue 7](#_Toc196562667)

[4. ATRIBUTOS DE CALIDAD DEL SOFTWARE 8](#_Toc196562668)

[Escenario de Funcionalidad 8](#_Toc196562669)

[Escenario de Usabilidad 8](#_Toc196562670)

[Escenario de confiabilidad 8](#_Toc196562671)

[Escenario de rendimiento 8](#_Toc196562672)

[Escenario de mantenibilidad 9](#_Toc196562673)

[Otros Escenarios 9](#_Toc196562674)

1. INTRODUCCIÓN
   1. Propósito (Diagrama 4+1)

La Aplicación para la Generación Automatizada de un Diccionario de Datos tiene como objetivo crear una herramienta que permita extraer, organizar, visualizar y generar un diccionario de datos de forma rápida y precisa. Facilitará la documentación y el análisis de bases de datos, ayudando a desarrolladores y administradores a crear diccionarios de forma eficiente, optimizando el proceso y reduciendo significativamente el tiempo de trabajo.

Desde una visión global de arquitectura basada en el modelo 4+1:

* Vista lógica: módulos que extraen, procesan y presentan los datos.
* Vista de procesos: prioriza eficiencia para procesar grandes bases de datos rápidamente.
* Vista de desarrollo: estructura modular y fácil de mantener.
* Vista física: se puede instalar en servidores locales o en la nube.
  1. Alcance

El sistema se encargará de:

* Extraer la estructura de la base de datos (tablas, columnas, tipos de datos, relaciones y restricciones).
* Generar una representación en formato exportable a Excel o PDF.
* Integrarse con bases de datos relacionales como SQL, con la posibilidad de expandirse a otros motores en el futuro.
* Permitir la personalización de las descripciones de los elementos del diccionario de datos.
* Brindar una interfaz intuitiva para la navegación y exportación de la información.
  1. Definición, siglas y abreviaturas
* Diccionario de Datos: Documento que describe la estructura de una base de datos, incluyendo sus entidades, atributos y relaciones.
* ERD (Entity-Relationship Diagram): Representación gráfica de las relaciones entre entidades en una base de datos.
* SGBD (Sistema de Gestión de Bases de Datos): Software que permite la creación, manipulación y administración de bases de datos.

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

El objetivo es Desarrollar una aplicación web que automatice la extracción, organización y generación de diccionarios de datos a partir de bases de datos relacionales, optimizando el proceso de documentación y facilitando su análisis y actualización.

* 1. Priorización de requerimientos

|  |  |  |  |
| --- | --- | --- | --- |
| **ID. Requerimiento** | **Nombre del Requisito** | **Descripción de Requisito** | **Prioridad** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Requerimientos Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **ID. Requerimiento** | **Nombre del Requisito** | **Descripción de Requisito** | **Prioridad** |
| RF-001 |  |  |  |
| RF-002 |  |  |  |
| RF-003 |  |  |  |
| RF-004 |  |  |  |
| RF-005 |  |  |  |

### Requerimientos No Funcionales – Atributos de Calidad

|  |  |  |  |
| --- | --- | --- | --- |
| **ID. Requerimiento** | **Nombre del Requisito** | **Descripción de Requisito** | **Prioridad** |
| RNF-001 |  |  |  |
| RNF-002 |  |  |  |
| RNF-003 |  |  |  |
| RNF-004 |  |  |  |
| RNF-005 |  |  |  |

* 1. Restricciones

Presupuestarias:

* El proyecto debe realizarse con un presupuesto limitado que cubra el desarrollo, las pruebas y la infraestructura mínima necesaria.
* Se priorizará el uso de tecnologías gratuitas o de código abierto para reducir costos.

Temporales:

* La aplicación debe estar lista en un plazo máximo de 6 meses, siguiendo un cronograma que incluya análisis, desarrollo, pruebas e implementación.
* No se pueden extender los plazos establecidos.

Tecnológicas:

* Se recomienda el uso de tecnologías accesibles y estables para asegurar la facilidad de mantenimiento y futuras mejoras.

Seguridad y Privacidad:

* Se deben proteger las credenciales de conexión a las bases de datos mediante mecanismos seguros.

Recursos Humanos:

* El equipo estará limitado a un pequeño grupo (2 a 3 personas), por lo que las herramientas elegidas deben ser fáciles de integrar y utilizar.
* La capacitación técnica deberá ser mínima para no retrasar el avance del proyecto.

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

* 1. Vista de Caso de uso

[En esta sección se describen los casos de uso del sistema (nombre de la aplicación), donde se abarcan todas las funcionalidades del sistema, se muestran los actores que interactúan en el sistema y las funcionalidades asociadas; asimismo se listará los casos de uso o escenarios del modelo de casos de uso que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que impliquen algún punto especialmente delicado de la arquitectura.

La documentación a incluir en esta sección corresponde a la obtenida como consecuencia de la actividad “Realización de casos de uso”:

- Flujos de eventos- Diseño: descripción textual de cómo se realiza el caso de uso en términos de los objetos que colaboran. Resumen de los diagramas conectados con el caso de uso y explicación de sus relaciones.

- Diagramas de interacción: Diagramas de secuencia, Diagramas de colaboración, objetos participantes, Diagramas de clases.

- Requisitos derivados: Descripción textual que recoge todos los requisitos, normalmente los no funcionales, de la realización del caso de uso no que han de tenerse en cuenta durante la implementación]

### Diagramas de Casos de uso

La descripción de la estructura se ilustra utilizando un conjunto de casos de uso escenarios lo que genera una nueva vista. Los escenarios describen secuencia de iteraciones entre objetos y entre procesos. Se utilizan para identificar y validar el diseño de arquitectura.

* 1. Vista Lógica

[La vista lógica se encarga de representar los requerimientos funcionales del sistema. Esta sección describe las partes del diseño del modelo significativas para la arquitectura, tales como subsistemas y paquetes.]

### 

### Diagrama de Subsistemas (paquetes)

[Diagrama que define los límites entre el sistema, o parte del sistema, y su ambiente, mostrando las entidades que interactúan con él. ​ Este diagrama es una vista de alto nivel de un sistema.

Asimismo, se debe desplegar las partes arquitectónicamente significativas del modelo de diseño, como ser la descomposición en capas, subsistemas o paquetes. Una vez presentadas estas unidades lógicas principales, se profundiza en ellas hasta el nivel que se considere adecuado.]

### Diagrama de Secuencia (vista de diseño)

### Diagrama de Colaboración (vista de diseño)

### Diagrama de Objetos

### Diagrama de Clases

### Diagrama de Base de datos (relacional o no relacional)

* 1. Vista de Implementación (vista de desarrollo)

[Se detalla la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación de manera más detallada]

### Diagrama de arquitectura software (paquetes)

[Se detalla la manera como fue implementado el sistema propuesto, se describe visualmente las capas que tiene el sistema, como están distribuidas y sus principales funciones]

### Diagrama de arquitectura del sistema (Diagrama de componentes)

[Se detalla la manera como fue implementado el sistema propuesto, se describe visualmente las capas que tiene el sistema, como están distribuidas y sus principales funciones]

* 1. Vista de procesos

[Describe la descomposición del sistema procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican.]

### Diagrama de Procesos del sistema (diagrama de actividad)

[Se realizará un diagrama del o los procesos del sistema donde se exponga las actividades donde interviene el sistema propuesto, adicionando diagramas que definan el detalle la descomposición del sistema en procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican]

* 1. Vista de Despliegue (vista física)

[Se despliega uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y hará el despliegue del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos]

### Diagrama de despliegue

[un diagrama de despliegue, amplía el sistema de software y muestra los contenedores (aplicaciones, almacenamiento de datos, microservicios, etc.) que componen este sistema de software]

# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

Para asegurar que la aplicación cumpla bien su propósito y sea práctica para los usuarios, se consideran los siguientes puntos importantes:

* Facilidad de uso: La aplicación tendrá un diseño simple y claro, para que cualquier usuario, aunque no tenga mucha experiencia, pueda generar su diccionario de datos sin problemas.
* Confiabilidad: El sistema debe ser seguro al momento de extraer la información de las bases de datos, generando diccionarios correctos y sin errores.
* Rapidez: La herramienta debe funcionar de forma ágil, especialmente al trabajar con bases de datos de tamaño pequeño o mediano, sin hacer esperar mucho al usuario.
* Capacidad de crecimiento: Aunque al inicio trabajará con SQL, debe poder ampliarse fácilmente para soportar otras bases de datos si se necesita en el futuro.
* Seguridad: La información sensible, como usuarios y contraseñas de las bases de datos, será protegida para evitar riesgos de accesos no autorizados.
* Facilidad de mantenimiento: El código y la estructura de la aplicación serán ordenados y documentados, facilitando que cualquier mejora o reparación futura se haga rápidamente.

Escenario de Funcionalidad

La aplicación debe ser capaz de extraer de manera automática los metadatos de diversas bases de datos (como MySQL y SQL Server) y generar un diccionario de datos completo y estructurado. El sistema debe permitir al usuario generar el diccionario de manera rápida y precisa, así como actualizarlo automáticamente cuando haya cambios en la estructura de la base de datos. La aplicación también debe permitir la exportación de los diccionarios de datos en formatos como JSON, XML o CSV.

Escenario de Usabilidad

La interfaz de la aplicación será fácil de usar y diseñada para diferentes tipos de usuarios, como desarrolladores, administradores de bases de datos y estudiantes. Se buscará que la generación del diccionario de datos sea sencilla, permitiendo a los usuarios interactuar de manera eficiente con el sistema. La navegación será clara, y los usuarios podrán seleccionar bases de datos, generar el diccionario, y acceder a la información de manera rápida y directa, sin necesidad de tener conocimientos técnicos avanzados en bases de datos.

Escenario de confiabilidad

Es fundamental que el sistema sea confiable. El diccionario de datos generado debe ser exacto, sin perder ni corromper la información, incluso cuando se trabajen con bases de datos grandes. La aplicación debe ser capaz de gestionar errores de manera efectiva, por ejemplo, si no puede acceder a la base de datos o si se presenta un error en el proceso de extracción de metadatos. Además, se implementarán medidas de seguridad como autenticación para garantizar que solo los usuarios autorizados puedan realizar cambios en los diccionarios de datos.

Escenario de rendimiento

El rendimiento de la aplicación es esencial para asegurar una experiencia fluida para el usuario. La aplicación debe ser capaz de generar y actualizar diccionarios de datos de manera eficiente, sin tiempos de espera largos, incluso cuando se trabaja con bases de datos de gran tamaño. El tiempo de respuesta debe ser rápido, garantizando que el sistema pueda manejar múltiples solicitudes sin degradar la experiencia del usuario.

Escenario de mantenibilidad

La aplicación debe estar diseñada de manera que sea fácil de mantener y actualizar en el futuro. Si se necesitan agregar nuevas funciones o mejorar las existentes, el sistema debe permitir hacerlo sin mayores complicaciones. Además, el código debe estar bien estructurado y documentado para que los desarrolladores puedan modificarlo o adaptarlo cuando sea necesario, sin afectar la estabilidad de la aplicación. Las actualizaciones automáticas del diccionario de datos también deben ser fácilmente configurables.

Otros Escenarios

La aplicación debe ser capaz de manejar grandes volúmenes de datos sin afectar el rendimiento. El tiempo requerido para generar el diccionario de datos debe ser razonable, incluso cuando se trabaje con bases de datos grandes o con estructuras complejas. Además, el sistema debe optimizar el uso de recursos, como el procesamiento y la memoria, para asegurar que el rendimiento se mantenga alto a medida que el número de bases de datos y los usuarios aumenten. Esto garantizará que la herramienta sea escalable y capaz de responder eficientemente a un mayor número de solicitudes y usuarios en el futuro.