****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Plataforma de Monitoreo y Análisis del Mercado Laboral en Ingeniería de Sistemas**

Curso: Inteligencia de Negocios

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

Daleska Nicolle Fernandez Villanueva (2021070308)

Andree Sebastian Flores Melendez (2017057494)

Mario Antonio Flores Ramos (2018000597)

**Tacna – Perú**

***2024***

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

Sistema Plataforma de Monitoreo y Análisis del Mercado Laboral en Ingeniería de Sistemas 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 |

INDICE GENERAL

Contenido

[1. INTRODUCCIÓN 5](#_Toc69808834)

[1.1. Propósito (Diagrama 4+1) 5](#_Toc69808835)

[1.2. Alcance 5](#_Toc69808836)

[1.3. Definición, siglas y abreviaturas 5](#_Toc69808837)

[1.4. Organización del documento 5](#_Toc69808838)

[2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS 5](#_Toc69808839)

[2.1.1. Requerimientos Funcionales 5](#_Toc69808840)

[2.1.2. Requerimientos No Funcionales – Atributos de Calidad 5](#_Toc69808841)

[3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 6](#_Toc69808842)

[3.1. Vista de Caso de uso 6](#_Toc69808843)

[3.1.1. Diagramas de Casos de uso 6](#_Toc69808844)

[3.2. Vista Lógica 6](#_Toc69808845)

[3.2.1. Diagrama de Subsistemas (paquetes) 7](#_Toc69808846)

[3.2.2. Diagrama de Secuencia (vista de diseño) 7](#_Toc69808847)

[3.2.3. Diagrama de Colaboración (vista de diseño) 7](#_Toc69808848)

[3.2.4. Diagrama de Objetos 7](#_Toc69808849)

[3.2.5. Diagrama de Clases 7](#_Toc69808850)

[3.2.6. Diagrama de Base de datos (relacional o no relacional) 7](#_Toc69808851)

[3.3. Vista de Implementación (vista de desarrollo) 7](#_Toc69808852)

[3.3.1. Diagrama de arquitectura software (paquetes) 7](#_Toc69808853)

[3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes) 7](#_Toc69808854)

[3.4. Vista de procesos 7](#_Toc69808855)

[3.4.1. Diagrama de Procesos del sistema (diagrama de actividad) 8](#_Toc69808856)

[3.5. Vista de Despliegue (vista física) 8](#_Toc69808857)

[3.5.1. Diagrama de despliegue 8](#_Toc69808858)

[4. ATRIBUTOS DE CALIDAD DEL SOFTWARE 8](#_Toc69808859)

[Escenario de Funcionalidad 8](#_Toc69808860)

[Escenario de Usabilidad 8](#_Toc69808861)

[Escenario de confiabilidad 9](#_Toc69808862)

[Escenario de rendimiento 9](#_Toc69808863)

[Escenario de mantenibilidad 9](#_Toc69808864)

[Otros Escenarios 9](#_Toc69808865)

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

*[Se presenta una visión global y resumida de la arquitectura del sistema y de los objetivos generales del diseño. Se describen las influencias con los requisitos funcionales y no funcionales del sistema y las decisiones y prioridades establecidas – eficiencia vs. Portabilidad, por ejemplo.]*

* 1. Alcance

*[El documento se centrará en el desarrollo de la vista lógica del framework. Se incluyen los aspectos fundamentales del resto de las vistas y se omiten aquellas que no se consideren pertinentes como ser el caso de la vista de procesos.]*

* 1. Definición, siglas y abreviaturas

*[Este apartado proporciona las definiciones de todos los términos, acrónimos y abreviaturas utilizadas a lo largo del documento y que permiten una interpretación correcta del mismo. Se han de incluir los términos técnicos, caso de uso por ejemplo, y los específicos del entorno del sistema, lector de bandas por ejemplo. Es conveniente ordenarlos alfabéticamente]*

* 1. Organización del documento

*[Aquí va la organización del proyecto]*

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

[Establezca las prioridades de los requerimientos y las restricciones del proyecto)

* 1. Priorización de requerimientos

*[Se procede a desplegar los requerimientos funcionales y no funcionales desde una perspectiva de priorización, mediante una tabla resumen donde pueda desplegar los requerimientos del sistema de la siguiente forma:*

|  |  |  |
| --- | --- | --- |
| *ID* | *Descripcion* | *Prioridad* |
|  |  |  |

*Asimismo con esta prioridad se definirá el orden de implementación.]*

### Requerimientos Funcionales

*[Definir la prioridad de los requerimientos funcionales.]*

|  |  |  |
| --- | --- | --- |
| *ID* | *Descripcion* | *Prioridad* |
|  |  |  |

### 

### Requerimientos No Funcionales – Atributos de Calidad

*[Definir la prioridad de los requerimientos NO funcionales.]*

|  |  |  |
| --- | --- | --- |
| *ID* | *Descripcion* | *Prioridad* |
|  |  |  |
|  |  |  |

*[Los Atributos de Calidad (QAs) son propiedades medibles y evaluables de un sistema, estas propiedades son usadas para indicar el grado en que el sistema satisface las necesidades de los stakeholders [Wojcik 2013].*

*Los QAs además son concebidos como aquellos requerimientos que no son funcionales. De hecho, la funcionalidad es mayormente ortogonal a los QAs; un diseño puede cumplir con la funcionalidad deseada y fallar a la hora de satisfacer sus requerimientos de calidad. De esta manera, se entiende a la funcionalidad como la capacidad del sistema para hacer el trabajo para el cual fue pensado, independientemente de la estructura. Existen QAs mayormente usados que se suelen identificar en numerosos sistemas y se tienen que describir, aunque la lista no es fina ya que muy a menudo hay situaciones en que podrían identificarse y proponerse nuevas propiedades para las diversas necesidades de stakeholders.]*

* 1. Restricciones

*[Aquí van las restricciones 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)

### Diagrama de despliegue

# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

*[Los Atributos de Calidad (QAs) son propiedades medibles y evaluables de un sistema, estas propiedades son usadas para indicar el grado en que el sistema satisface las necesidades de los stakeholders [Wojcik 2013].*

*Los QAs además son concebidos como aquellos requerimientos que no son funcionales. De hecho, la funcionalidad es mayormente ortogonal a los QAs; un diseño puede cumplir con la funcionalidad deseada y fallar a la hora de satisfacer sus requerimientos de calidad. De esta manera, se entiende a la funcionalidad como la capacidad del sistema para hacer el trabajo para el cual fue pensado, independientemente de la estructura. Existen QAs mayormente usados que se suelen identificar en numerosos sistemas y se tienen que describir, aunque la lista no es fina ya que muy a menudo hay situaciones en que podrían identificarse y proponerse nuevas propiedades para las diversas necesidades de stakeholders.]*

*.*

Escenario de Funcionalidad

*[se califica de acuerdo con el conjunto de características y capacidades del programa, la generalidad de las funciones que se entregan y la seguridad general del sistema.]*

Escenario de Usabilidad

*[Este atributo de calidad se refiere a la facilidad con la que un usuario puede aprender a utilizar e interpretar los resultados producidos por un sistema [Barbacci 1995]. Para este atributo de calidad, se suelen considerar diversos aspectos de la interacción humano computadora, tales como: aprendizaje del sistema, utilización eficiente del sistema, minimización del impacto de errores, adaptación del sistema a las necesidades del usuario, confianza y satisfacción, entre otros.]*

Escenario de confiabilidad

*[Es el equilibrio entre la confidencialidad, la integridad, la irrefutabilidad y la disponibilidad de la información y datos manipulados por el sistema. Se trata del estado de un sistema, el cual puede ser transitorio y volátil. La seguridad de un sistema se caracteriza por mecanismos y técnicas empleados para intentar reducir los más posible el impacto provocado por un ataque, y las amenazas (entendidas como los caminos mediante los cuales se pueden provocar un ataque).*

*Abarca los planos de observación físico, lógico y humanos. Posee tres tipos de enfoque: prevención, precaución y reacción.]*

Escenario de rendimiento

*[Se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto y la eficiencia.] (Pressman 2010, pág. 187)*

Escenario de mantenibilidad

*[Combina la capacidad del programa para ser ampliable (extensibilidad), adaptable y servicial. (Pressman 2010, pág. 187)*

Otros Escenarios

*[“Otros escenarios como por ejemplo: Performance”*

***Performance****: El atributo de calidad Performance se refiere a la capacidad de responder, ya sea el tiempo requerido para responder a eventos determinados, o bien, la cantidad de eventos procesados en un intervalo de tiempo dado. La Performance caracteriza la proyección en el tiempo de los servicios entregados por el sistema.]*