**Registro de Nombres**

**Documento de arquitectura de software**

**Versión <1.0>**

**Índice Analítico**

1. Introducción 4

1.1 Meta 4

1.2 Alcance 4

1.3 Definiciones, Acrónimos y Abreviaturas 4

1.4 Referencias 4

1.5 Visión general 4

2. Representación Arquitectónica 4

3. Objetivos y limitaciones de la arquitectura 4

4. Vista de caso de uso 5

4.1 Realizaciones de casos de uso 5

5. Visión lógica 5

5.1 Visión general 5

5.2 Paquetes de diseño arquitectónicamente significativos 5

6. Visión de proceso 5

7. Visión de Implementación 5

8. Visión de Implementación 5

8.1 Visión general 5

8.2 capas 6

9. Vista de datos (opcional) 6

10 Tamaño y rendimiento 6

11 Calidad 6

**Documento de arquitectura de software**

1. **Introducción**

*[La introducción de****Documento de arquitectura de software*** *proporciona una visión general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias y descripción general del****Documento de arquitectura de software****.]*

* 1. **Meta**

Este documento proporciona una descripción general completa de la arquitectura del sistema, utilizando varias vistas arquitectónicas para representar diferentes aspectos del sistema. El propósito de este documento es capturar y comunicar las decisiones arquitectónicas importantes que se tomaron con respecto al sistema.

*[Esta sección define el rol o propósito del****Documento de arquitectura de software****, en la documentación del proyecto en su conjunto, y describe brevemente la estructura del documento. Se identifica la audiencia específica del documento, con una indicación de cómo esperan utilizar el documento.]*

Este documento tiene como objetivo principal describir la estructura, los componentes y el funcionamiento del sistema de gestión de alumnos diseñado para la escuela EBAC. El sistema ha sido concebido para brindar una herramienta eficiente y escalable para la gestión de datos de estudiantes, desde el ingreso de información básica hasta la eliminación de registros.

El sistema cubre las funcionalidades de registro, actualización, consulta y eliminación de datos académicos de los estudiantes de EBAC.

* 1. **Alcance**

*[Una breve descripción de la utilidad del Documento de arquitectura de software, lo que se ve afectado o influenciado por ese documento.]*

El Documento de Arquitectura de Software es como un mapa detallado que guía la construcción y el mantenimiento de un sistema. Sirve para:

* **Comprender el sistema:** Proporciona una visión global y detallada de cómo está construido el sistema, facilitando su entendimiento tanto para los desarrolladores actuales como para los nuevos que se unan al proyecto.
* **Tomar decisiones:** Ayuda a tomar decisiones informadas sobre cambios y actualizaciones, al mostrar las implicaciones de cada modificación en el sistema completo.
* **Comunicación efectiva:** Facilita la comunicación entre diferentes equipos (desarrolladores, testers, stakeholders) al proporcionar un lenguaje común y un punto de referencia.
* **Evitar errores:** Ayuda a prevenir errores y a detectar posibles problemas de diseño antes de que se conviertan en problemas mayores.
* **Asegurar la calidad:** Contribuye a garantizar la calidad del software al establecer estándares y buenas prácticas.
* **Facilitar el mantenimiento:** Hace que el sistema sea más fácil de mantener y actualizar a largo plazo.
  1. **Definiciones, Acrónimos y Abreviaturas**

*[Esta subsección contiene las definiciones de todos los términos, siglas y abreviaturas necesarias para interpretar correctamente el****Documento de arquitectura de software****. Esta información se puede proporcionar consultando el Glosario del proyecto.]*

* 1. **Visión general**

*[Esta subsección describe lo que el resto de los****Documento de arquitectura de software*** *contiene y explica cómo está organizado el documento.]*

1. **Representación Arquitectónica**

*[Esta sección describe cuál es la arquitectura de software del sistema actual y cómo se representa. Desde el****Vista de caso de uso****,****Visión******Lógica****,****Visión de proceso****,****Vista de******Implantación*** *Es****Vista de******Implementación****, enumera las vistas requeridas y, para cada vista, explica qué tipos de elementos de modelo contiene.]*

1. **Objetivos y limitaciones de la arquitectura**

*[Esta sección describe los requisitos y objetivos del software que tienen algún impacto en la arquitectura; por ejemplo, seguridad, garantía, privacidad, uso de un producto listo para usar desarrollado internamente, portabilidad, distribución y reutilización. También captura las restricciones especiales que pueden aplicarse: estrategia de diseño e implementación, herramientas de desarrollo, estructura del equipo, cronograma, código fuente heredado, etc.]*

1. **Vista de caso de uso**

*[Esta sección enumera casos de uso o escenarios de modelos de casos de uso cuando representan una funcionalidad central y significativa del sistema final o, cuando tienen una gran huella arquitectónica: experimentan muchos elementos arquitectónicos o cuando enfatizan o ilustran un punto complejo y específico de la arquitectura.]*

* 1. **Realizaciones de casos de uso**

*[Esta sección ilustra cómo funciona el software, presenta algunas realizaciones (o escenarios) de casos de uso seleccionados y explica cómo los diversos elementos del modelo de diseño contribuyen a su funcionalidad.]*

1. **Visión lógica**

*[Esta sección describe las partes arquitectónicamente significativas del modelo de diseño, como su división en subsistemas y paquetes. Además, para cada paquete significativo, muestra su desglose en clases y utilidades de clase. Presente las clases arquitectónicamente significativas y describa sus responsabilidades, así como algunas relaciones, operaciones y atributos importantes.]*

* 1. **Visión general**

*[Esta subsección describe la descomposición completa del modelo de diseño en términos de capas y la jerarquía del paquete.]*

* 1. **Paquetes de diseño arquitectónicamente significativos**

*[Para cada paquete importante, incluya una subsección con su nombre, una breve descripción y un diagrama de todos los paquetes y clases importantes que contiene.*

*Para cada clase significativa en el paquete, incluya su nombre, una breve descripción y, opcionalmente, una descripción de algunas de sus responsabilidades, operaciones y atributos clave.]*

1. **Visión de proceso**

*[Esta sección describe la descomposición del sistema en procesos livianos (hilos únicos de control) y procesos pesados ​​(grupos de procesos livianos). Organice la sección en grupos de procesos que se comunican o interactúan. Describa los principales modos de comunicación entre procesos, como el paso de mensajes y las interrupciones.]*

1. **Visión de Implementación**

*[Esta sección describe una o más configuraciones de la red física (hardware) en la que se implementa y ejecuta el software. Es una vista del modelo de implementación. Como mínimo, para cada configuración, debe indicar los nodos físicos (computadoras, CPU) que ejecutan el software y sus interconexiones (bus, LAN, punto a punto, etc.).****Visión de proceso*** *en los nodos físicos.]*

1. **Visión de Implementación**

*[Esta sección describe la estructura general del modelo de implementación, la división del software en capas y subsistemas en el modelo de implementación y todos los componentes arquitectónicos significativos.]*

* 1. **Visión general**

*[Esta subsección nombra y define las diversas capas y su contenido, las reglas que determinan la inclusión en una capa en particular y los límites entre las capas. Incluya un diagrama de componentes que muestre las relaciones entre las capas. ]*

* 1. **capas**

*[Para cada capa, incluya una subsección con su nombre, una lista de subsistemas ubicados en la capa y un diagrama de componentes.]*

1. **Vista de datos (opcional)**

*[Una descripción de la perspectiva de almacenamiento de datos persistentes del sistema. Esta sección es opcional si los datos persistentes son escasos o inexistentes, o si la conversión entre el modelo de diseño y el modelo de datos es trivial.]*

1. **Tamaño y rendimiento**

*[Una descripción de las características clave de dimensionamiento del software que tienen un impacto en la arquitectura, así como las limitaciones en el rendimiento deseado.]*

1. **Calidad**

*[Una descripción de cómo la arquitectura del software contribuye a todas las funciones (excepto la funcionalidad) del sistema: extensibilidad, confiabilidad, portabilidad, etc. Si estas características tienen un significado especial, como implicaciones de seguridad, garantía o privacidad, deben estar claramente delineadas.]*