|  |
| --- |
| Documento de Análisis de Diagnostico |
| **SISACAD – SISTEMA ACADEMICO** |

Historial de Versiones

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| version | DESCRIPCION | AUTOR | FECHA CREACION | APROBADO POR | FECHA APROBACION |
| 1.0 | Análisis de diagnostico | Juan Carlos de la Cruz | 25/07/2024 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contenido

[1. Alcance del Documento Análisis de Diagnostico Situacional SISACAD – Sistema Academico UNAT 4](#_Toc173305887)

[2. Situación Actual 4](#_Toc173305888)

[3. Análisis de Impacto Reemplazo PAN 5](#_Toc173305889)

[3.1. Diagrama de Arquitectura 5](#_Toc173305890)

[3.1.1. Diagrama de Infraestructura 5](#_Toc173305891)

[3.1.2. Diagrama de flujo de uso. 6](#_Toc173305892)

[3.1.3. Diagrama de flujo de transmisión de datos 7](#_Toc173305893)

[3.2. Generalidades de Componentes 7](#_Toc173305894)

[3.3. Recomendaciones Generales 14](#_Toc173305895)

[3.4. Diagnostico Final del SISACAD 14](#_Toc173305896)

[3.5. Análisis de implementación de reemplazo del SISACAD 14](#_Toc173305897)

# Alcance del Documento Análisis de Diagnostico Situacional SISACAD – Sistema Academico UNAT

El alcance de este documento es documentar todos los componentes del SISACAD y evaluar la factibilidad de implementar las alternativas de reemplazo indicadas en la sección 3.5 de este documento. Todo esto con la finalidad de no afectar la operatividad de la UNAT.

# Situación Actual

*Tecnología:*

* *Tipo de aplicación: Aplicación de Web*
* *Base de datos: Mysql*
* *Lenguaje de programación: PHP v.5.2*
* *Servidor de aplicaciones: Apache*

*Funcionalidades:*

* *Administración*
  + *Principal*
  + *Adm. General*
  + *Cronograma*
  + *Datos alumno*
  + *Datos docente*
  + *Adm. Usuarios*
* *Designación de asignatura*
  + *Prog. de Cursos*
  + *Prog. de Horarios*
  + *Migrar Moodle*
  + *Reprogramacion*
  + *Ex.Complementario*
* *Matricula*
  + *Cert. de Estudios*
  + *Homologación*
  + *Pre / Matrícula*
  + *Habilitar*
* *Reporte*
  + *Reporte de actas*
  + *Reportes curriculares*
  + *Reportes estadísticos*
  + *Reportes estudiantes*
* *Seguimiento de Agresados*
  + *Calculo egresados*
  + *Reporte egresados*
* *Servicios*
  + *Constancia Estudio*
  + *Primera matricula*
  + *Cons. egresado*
  + *Retiros*
  + *Suspensiones*
  + *Expulsiones*
  + *Reservas*
  + *Renudacion*
  + *Cambiar Seccion*
  + *Reportes lote*
  + *Enc. Minedu*
  + *Exportar Moodle*
* *Gestion*
  + *Becas*
  + *Discapacitados*
  + *Bloqueos*
* *Indicadores*
  + *Semestrales*
* *Formatos*
  + *Siries*
  + *Siu*
  + *Boletín*

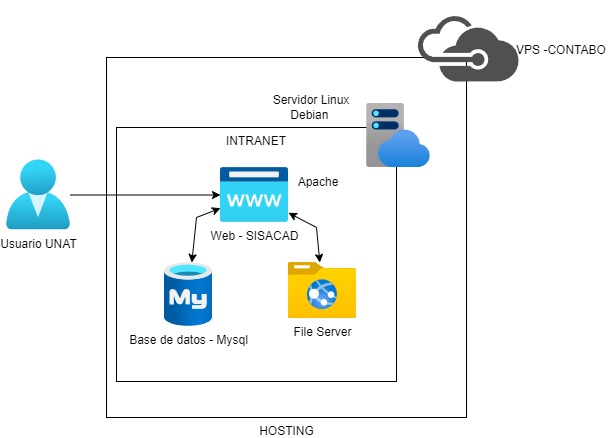
*Dependencias del Consolidados*

* *Moodle UNAT.*

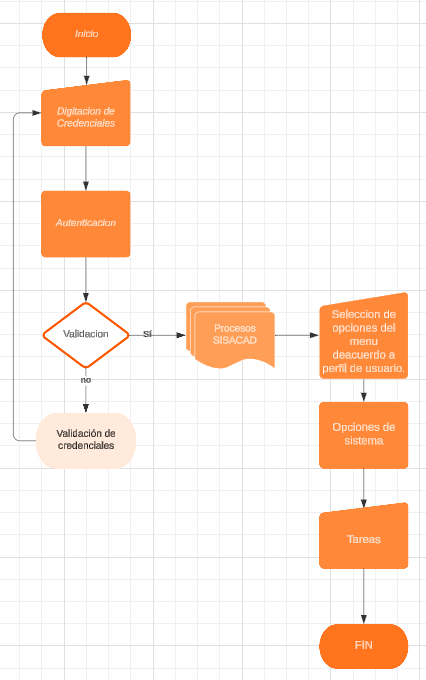
# Análisis de Impacto Reemplazo PAN

## Diagrama de Arquitectura

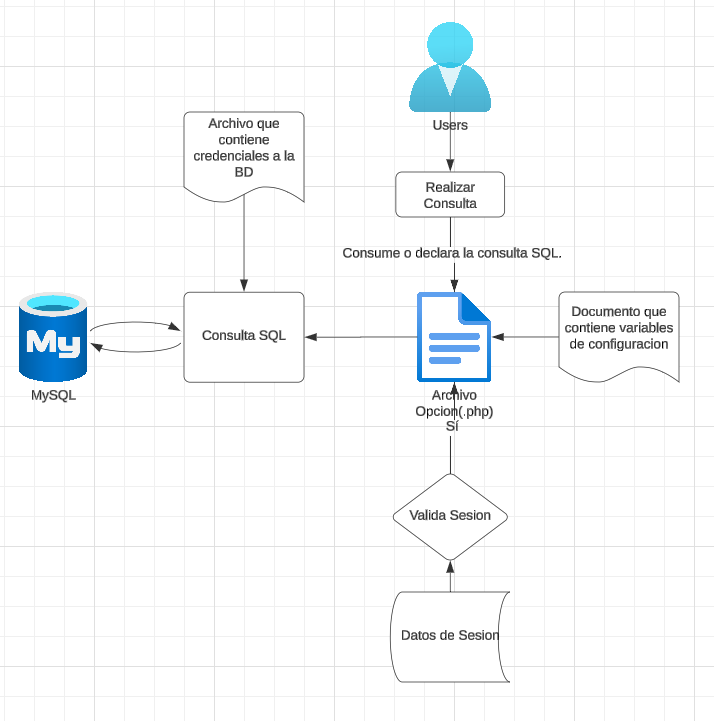
### Diagrama de Infraestructura



### Diagrama de flujo de uso.



### Diagrama de flujo de transmisión de datos



## Generalidades de Componentes

1. Estructura de Base de Datos:

*La base de datos no se encuentra debidamente normalizada.*

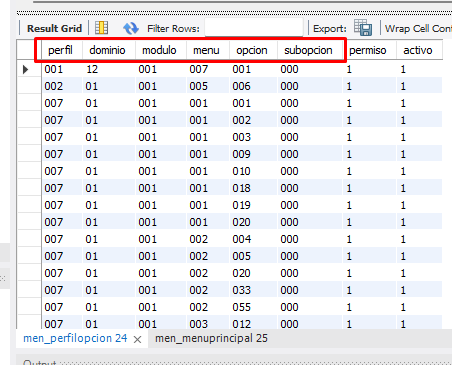
* *Se utiliza nomenclatura para nombrar tablas.*

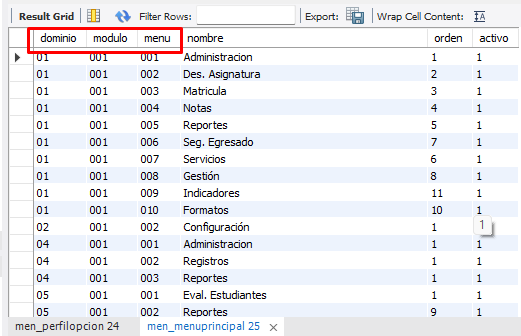


* *No utiliza llaves primarias, ni llaves foráneas.*



* *No contiene diccionario de datos.*
* *Duplicidad de datos*





* *Procedimientos almacenados con tablas temporales – existe la probabilidad de interferencia de datos con múltiples usuarios.*



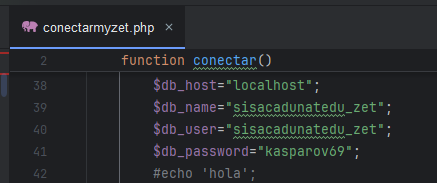
*Recomendaciones:*

* Normalizar la base de datos.
* Evitar el uso de tablas temporales aplicando estrategias propias del SQL.
* Nomenclatura estándar para los objetos.

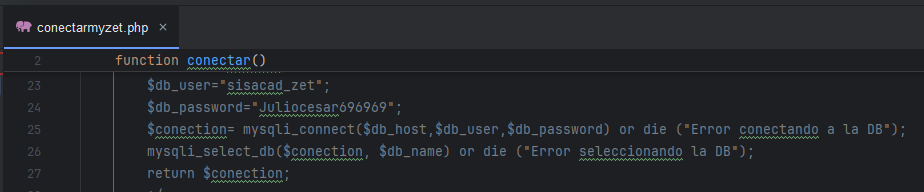
1. Estructura de archivos de configuración

*Detalles:*

* *Los archivos de configuración contienen información sensible en texto plano.*



* *Texto comentado*



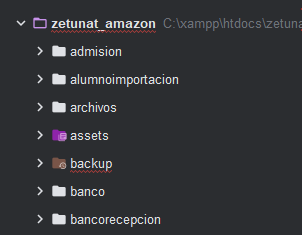
*Recomendaciones:*

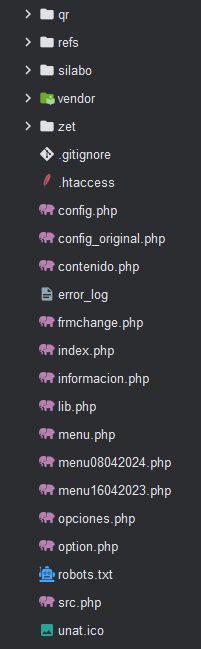
* *Utilizar archivos de configuración únicos y encriptados.*
* *Evitar tener código comentado.*

1. Estructura de archivos:

Detalles:

* No hay una adecuada estructura de carpetas en el proyecto.





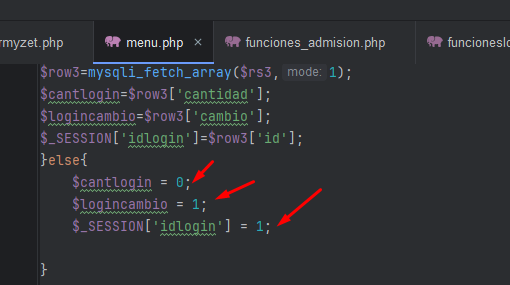
Recomendaciones:

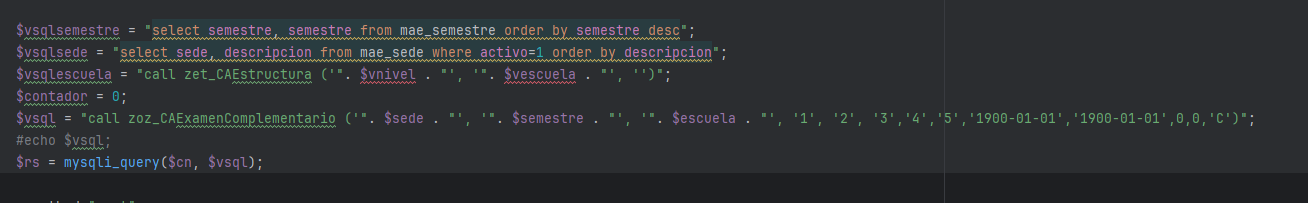
* Utilizar el Vertical Slicing (Clean Arquitecture)

1. Estructura de código:

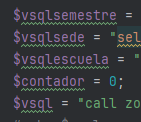
Detalles:

* Utilización de código duro.

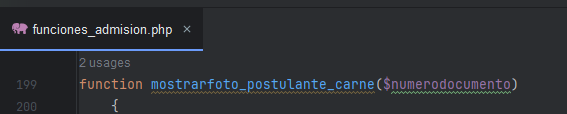


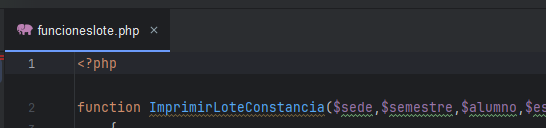


* No tienen estándares de programación.
  + Nombre de variables



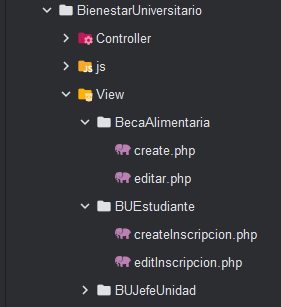
* + Nombre de funciones (snake case, camel case)



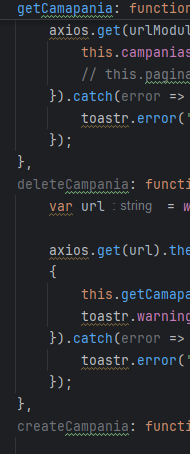


Recomendaciones:

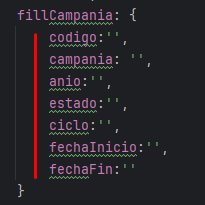
* Vertical Slicing



* Correcta utilización de camel case



* Nombrar variables claras y definidas en un ámbito exclusivo.



## Recomendaciones Generales

El SISACAD es un software tipo **LEGACY** con las siguientes características:

* Obsolescencia Tecnológica
  + Tecnologías Antiguas
* Falta de Soporte y Actualizaciones
  + Soporte Limitado
* Difícil Mantenimiento
  + Código Antiguo
* Problemas de Compatibilidad
  + Integración Difícil
* Seguridad
  + Vulnerabilidades
* Rendimiento y Escalabilidad
  + Limitaciones
* Costos de Mantenimiento
  + Costoso de Mantener
* Dificultad para Implementar Nuevas Funcionalidades
  + Flexibilidad Limitada
* Dependencia de Personal Especializado
  + Conocimiento Específico
* Falta de Documentación
  + Documentación Insuficiente
* Problemas de Usabilidad
  + Experiencia de Usuario
* Migración Compleja
  + Transición Dificultosa

## Diagnostico Final del SISACAD

En resumen, el SISACAD sigue funcional y valioso, sus desventajas pueden hacer que sea necesario considerar una modernización o reemplazo eventual para mantener la competitividad y la eficiencia en el entorno tecnológico actual.

## Análisis de implementación de reemplazo del SISACAD

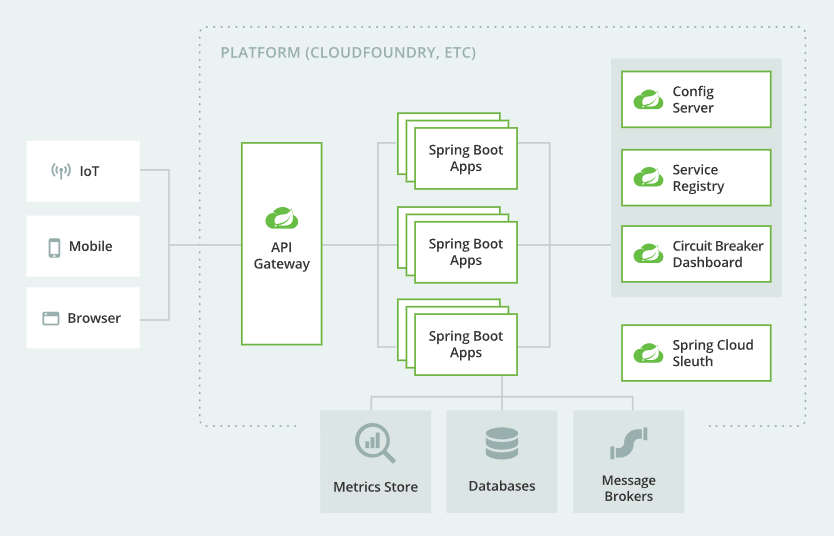
Debido a la complejidad y el impacto en la operatividad del negocio se propone:

**Solución Personalizada**

* **Descripción**: Desarrollar una solución a medida que se adapte específicamente a las necesidades y procesos del negocio, a menudo utilizando tecnologías modernas.
* **Ventajas**:
  + A medida para ajustarse a los requisitos específicos del negocio.
  + Mayor control sobre el diseño y las funcionalidades.

**Propuesta Tecnológica**

Un nuevo software con una arquitectura en microservicios, estos permitiran agregar funcionalidades de manera optima.



Al desarrollar una solución a medida que se adapte específicamente a las necesidades y procesos del negocio, recomendamos utilizar:

* **Lenguajes de Programación**
  + Propuesta 1: Net Core 8.0
  + Propuesta 2: PHP 8.3
  + Propuesta 3: Java 21
* **Frameworks de Desarrollo**
  + Propuesta 1: Net Core 8.0
  + Propuesta 2: Framework Laravel 11.x
  + Propuesta 3: Spring Boot
* **Frameworks de Desarrollo Frontend**
  + Propuesta 1: Angular 17
  + Propuesta 2: React 19
* **Bases de Datos**
  + Propuesta 1: SQL SERVER 2022
  + Propuesta 2: MariaDB
* **Infraestructura y DevOps**
  + Propuesta 1: Docker
  + Propuesta 2: Kubernetes
* **Automatización y CI/CD**:
  + Propuesta 1: Jenkins
  + Propuesta 2: GitLab
* **Control de Versiones**
  + Propuesta 1: Git
  + Propuesta 2: Team Foundation
* **Arquitectura y Diseño**
  + Propuesta 1: Microservicios
  + Propuesta 2: Serveless
* **Seguridad**
  + Propuesta 1: JWT
  + Propuesta 2: OAuth 2
* **QA**
  + Propuesta 1: Pruebas Unitarias
* **Mobile**
  + Propuesta 1: React Native
  + Propuesta 2: Kotlin Multiplatform

**Detalles:**

Esta propuesta no impactará el funcionamiento actual del SISACAD dado que la actualización se realizará por módulos y estos seguirán intercomunicándose.

No se perderá funcionalidad del SISACAD hasta que el software haya sido reemplazado totalmente.

Utilizando una Metodología de desarrollo Waterfall.

