

DOCUMENTO DE ARQUITECTURA DEL SOFTWARE

Viernes, 13 de marzo de 2020

Proyecto: SIS-QSF

30 de Enero del 2020

Colaboradores:

Briyant Iván Zeuz Pérez Reyes

José Luis González Ruiz

Christian Benigno Morales Morales

Luis Alberto Ignacio Esteban

Contenido

[1. Introducción 3](#_Toc35177160)

[1.1 Alcance. 3](#_Toc35177161)

[2. Resumen Arquitectónico. 3](#_Toc35177162)

[2.1 Hechos mas importantes. 3](#_Toc35177163)

# Introducción

## Alcance.

El SIS-QSF tiene como finalidad principal administrar y gestionar las quejas, sugerencias y felicitaciones emitidas en el Instituto Tecnológico Superior del Sur de Guanajuato que contribuya en la verificación del cumplimiento de los servicios ofrecidos a los estudiantes durante su formación profesional.

# Resumen Arquitectónico.

## Hechos mas importantes.

Mediante este documento son especificados las tecnologías y componentes de alto nivel que se van a implementar en el desarrollo del SIS-QSF con el que el equipo de trabajo se centrara en utilizar durante todo su proceso de desarrollo.

El SIS-QSF esta visualizado como una herramienta que reemplace el proceso de emisión de alguna queja o sugerencia emitida por algún alumno o gente externa (padres de familia,etc) en su forma tradicional en papel por un sistema automatizado que ayude de manera eficiente a realizar las mismas funciones mediante la implementación y uso de TICs.

Algunas de las funciones principales del sistema a manera de lista:

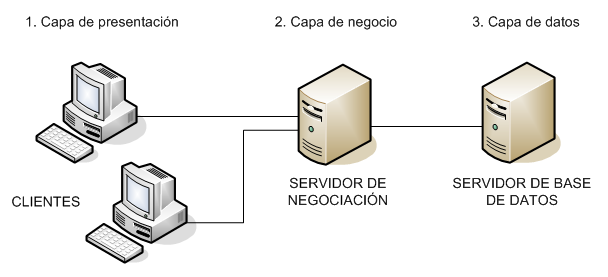
* Administración individual de la QSF.
  + Establecer una prioridad a la QSF.
  + Redirigir la solicitud.
  + Visualización de las QSF.
* Almacenamiento de información.

Los datos ingresados previamente se guardarán en una BD para su posterior utilización.

* Generación de reportes.
* Generar o emitir una QSF.
* Informar los términos y condiciones a los que están sujetas las QSF.
* Filtrado y búsquedas en el buzon.

## Estilo arquitectónico.

Para el desarrollo de este proyecto se ha decidido utilizar un modelo tradicional de 3 capas: Capa de presentación, modelo y datos.



Cada capa contiene varios componentes que en conjunto darán solución al sistema.

## Objetivos de la arquitectura.

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de calidad | Rango Requerido | Motivo | Estrategias para lograr el objetivo |
| Agilidad | Alto | Se espera que el producto pueda ser cambiado rápidamente bajo peticiones específicas del usuario. | * Apegarse a los estándares de calidad del Sistema de Gestión de Calidad de la institución. * Aplicar en lo posible los patrones de alta cohesión y bajo acoplamiento. * Uso de un lenguaje de desarrollo tal como C# y ASP.net para personal que se integre a realizar algún cambio en el diseño. |
| Interoperabilidad | Medio-Alto | El software contiene componentes que deben ser capaces de intercambiar información en más de una forma (síncrono y asíncrono). | * Utilizar comunicación entre el servidor WEB y de datos . * archivos de respaldo de base de datos. |
| Mantenimiento | Medio | El sistema debe tener facilitar la identificación de fallos, así como su recuperación tras los mismos. | * Utilizar un mecanismo de registro de defectos tipo log en cada una de las excepciones que genere el sistema. * Aplicar las mismas estrategias que para Agilidad. |
| Rendimiento | Alto | Se espera que el sistema mantenga una tasa de respuesta inferior al medio segundo en el 90% de sus operaciones. | * Integrar un SGBD con un tiempo aceptable de latencia. * Realizar pruebas de rendimiento para contabilizar los tiempos de respuesta. * Optimizar mediante revisiones entre colegas las transacciones que impliquen un procesamiento de datos que sea considerado grande. |
| Fiabilidad | Medio | Se espera que el sistema mantenga un rango de fiabilidad de seis a doce meses hasta requerir mantenimiento. | * Analizar en pares los procesos que requieran mecanismos de recuperación y mantenimiento preventivo para aplicarlos conforme corresponda durante la construcción iterativa del sistema, pudiendo generar esto flujos alternativos de recuperación en los casos de uso e inclusive nuevas funcionalidades. |
| Reutilización | Medio | Es de esperarse que los módulos del sistema sean reutilizables sirviendo como base durante la implantación del sistema en cualquier otro negocio. | * Utilizar el patrón módulos de enchufe (plug-in). |
| Escalabilidad | Medio-Alto | Se espera que no exista disminución en el rendimiento del sistema en cuanto al aumento de las unidades de punto de venta conectadas al sistema central. | * Realizar pruebas de rendimiento para contabilizar los tiempos de respuesta con base en la carga de terminales del sistema. |
| Seguridad | Medio | Es de esperarse un nivel de seguridad promedio en cuanto al acceso al sistema en el entorno de producción. Y un nivel promedio en comunicación a través de la red. | * Intentar encriptar los de datos en la comunicación por la red. * Utilizar encriptación de datos mediante almacenamiento en archivos comprimidos con contraseña para comunicación mediante archivos. |
| Compatibilidad | Medio-Alto | Se espera que el sistema de software funcione sobre ambientes de sistema operativo de última generación y posteriores. | * Utilizar las tecnologías lo más reciente posible y que sean ya estables tales como .Net framework 3.5. |
| Capacidad de prueba (Testability) | Alto | Es de esperarse que los componentes de software sean sometidos a pruebas que garanticen su calidad durante el proceso de desarrollo. | * Diseñar pruebas unitarias para cada módulo, mediante una herramienta auxiliar al IDE que permita que las pruebas puedan ser ejecutadas repetidas veces y que estas garanticen el adecuado funcionamiento de los módulos. |
| Usabilidad | Alto | Será requerido que el sistema tenga un alto nivel de usabilidad facilitando que pueda aprender a operar, preparar los insumos para, e interpretar los resultados de cada componente del sistema. | * Someter el software a pruebas de usabilidad. |