

Gestión de Proyectos de Software

Proyecto

Sistema de tickets

Integrantes:

José Luis Vázquez Corona

Daniel Manuel Lopez Almeida

Profesor: Ing. Ernesto Alejandro Lima

Camino Arenero No. 1101 Col. El Bajío

45019 Zapopan, Jalisco, México

***Capítulo I Planeación***

**1.1 Planteamiento del problema:**

Dentro de una organización el personal desempeña diferentes actividades y en ocasiones surgen algunos problemas que les impiden poder continuar con su trabajo de forma habitual tales como: problemas con software, virus en ordenadores, impresoras, etc. para lo cual se necesita avisar al departamento de sistemas para que puedan solucionar esas situaciones.

La comunicación es escasa entre usuarios y el departamento de sistemas para notificar los inconvenientes presentados o la demora en la atención de los problemas con mayor importancia puede crear pérdidas importantes a la organización de tiempo y dinero.

**1.2** **Objetivo**

Diseñar interfaces para los usuarios según sus roles en la organización, para que sirvan como medio de enlace entre el Departamento de Sistemas y estos, para poder dar seguimiento a los tickets y de esta manera resolver incidentes de una forma más óptima.

**1.3 Alcance**

Este proyecto será integrador el cual se realizara en diferentes bloques, el bloque en el cual se trabajara este semestre es diseñar la interfaz de usuario para la creación de un ticket de incidente y la generación de un id único para poder dar seguimiento del mismo.

**1.4 Justificación**

El propósito de la creación de este sistema de tickets, es llevar un control de las solicitudes de servicio de varias maneras entre las cuales están: prioridad, antigüedad y departamento. Lo anterior para satisfacer la atención a las necesidades a usuarios, con la mejor calidad.

**1.5 Antecedentes**

En la actualidad existe software de seguimiento de Incidentes también llamados Helpdesk el cual es un paquete de software que administra y mantiene listas de historial de situaciones, conforme son requeridos por una institución que las mismas generan, pero este tipo de software requiere un costo de licencia mensual por equipo (pc) más un costo adicional por instalación y soporte, tales como: ARTOLOGIK, JIRA, ZENDESK INTEGRIA IMS, los cuales son paquetes de software que necesitan ser utilizados por personas con experiencia en los mismos debido a que tienen funciones un poco intuitivas.

**1.6 Puntos de Función**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Complejidad | Baja | Media | Ala | oal |
| EI Enrada Inerna | 15 x 3 PF |  |  | 45 PF |
| EO Enrada exerna | 16 x 4 PF |  |  | 64 PF |
| EQ Consula exerna | 4 x 3 PF |  |  | 12 PF |
| ILF Archivo Logico Inerno | 4 x 7 PF |  |  | 28 |
| EIF ArchivoInerfaz exerno | 0 |  |  | 0 |
| PFSA | 149 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requisitos | | | |
| Comunicación de Datos | | | 3 |
| Procesamiento Distribuido | | | 1 |
| Objetivos de Rendimiento | | | 3 |
| Configuración del Equipamiento | | | 3 |
| Tasa de Transacciones | | | 3 |
| Entrada de Datos en línea | | | 4 |
| Interface Visual Usuario | | | 5 |
| Actualizaciones en línea | | | 4 |
| Procesamiento complejo | | | 3 |
| Reusabilidad del código | | | 0 |
| Facilidad de implementación | | | 3 |
| Facilidad de Operación | | | 3 |
| Instalaciones Múltiples | | | 3 |
| Facilidad de cambios | | | 3 |
| **Factor de Ajuste ( FA)** | | | 41 |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| PFA = PFSA \* [0.65 + (0.01 \* factor de ajuste)] | | | |
| **PFA** | 149,41 | | |
|  |  |  |  |
| KLOC = PFA \* Líneas de código por lenguaje | | | |
| **KLOC** | 4,78112 | | |

**1.7 Calculo de los costos y beneficios**

**El modo fue organico.**

|  |  |  |  |
| --- | --- | --- | --- |
| **COCOMO** | | | |
| Esfuerzo = 2.4 (KLOC)^1.05 | | | |
| **Esfuerzo** | 12,40844523 | | |
|  |  |  |  |
| Tiempo de Desarrollo = 2.5(Esfuerzo)^0.38 | | | |
| **Desarrollo** | 6,509574602 | | |
|  |  |  |  |
| Número de Personas = Esfuerzo / Desarrollo | | | |
| **Personas** | 1,906183735 | | |

|  |  |
| --- | --- |
| Costo = 18000 \* Personas | |
| **Costo** | $34.311,31 |

**1.8 Requerimientos Funcionales**

|  |  |  |
| --- | --- | --- |
| RF-Tickets-01 | Validar la inserción de nuevos contactos. | |
| Objetivos Asociados: | Se generará un algoritmo que le permita a la aplicación crear una lista de contactos y posteriormente insertará los nuevos contactos en la BD. | |
|
| Requisitos Asociados: |  | |
|
| Descripción: | Con una reg-ex permitirá saber si el correo cumple con los requerimientos básicos ID@servername.com. | |
|
| Pre-Condición: | Que el destinatario cuente con correo electrónico. | |
|
| Secuencia: | Paso: | Acción: |
| 1 | Obtener nombre, usuario, password y correo del contacto. |
| 2 | Escribir los datos del contacto. |
| 3 | Insertar el contacto a la BD. |
| Post-Condición: | Que no sea un usuario anteriormente registrado. | |
|
| Excepciones: |  | |
|  | |

|  |  |  |
| --- | --- | --- |
| RF-Tickets-02 | Validar la inserción de nuevos departamentos. | |
| Objetivos Asociados: |  | |
|
| Requisitos Asociados: |  | |
|
| Descripción: | Capturar el nombre de los departamentos con que cuenta la empresa. | |
|
| Pre-Condición: | Dependiendo del departamento se obtendrá la prioridad de respuesta. | |
|
| Secuencia: | Paso: | Acción: |
| 1 | Obtener el nombre del departamento y asignar prioridad. |
| 2 | Escribir los datos del departamento. |
| 3 | Insertar el departamento a la BD. |
| Post-Condición: | Las prioridades en departamentos pueden repetirse. | |
|
| Excepciones: |  | |
|  | |

|  |  |  |
| --- | --- | --- |
| RF-Tickets-03 | Validar la inserción de nuevos tickets. | |
| Objetivos Asociados: |  | |
|
| Requisitos Asociados: |  | |
|
| Descripción: | Verificar que todos los campos en la interfaz de tickets sean válidos. | |
|
| Pre-Condición: | Todos los campos necesitan estar completos. | |
|
| Secuencia: | Paso: | Acción: |
| 1 | Obtener: título, categoría, departamento y descripción. |
| 2 | Escribir los datos del ticket. |
| 3 | Insertar el ticket a la BD. |
| Post-Condición: |  | |
|
| Excepciones: |  | |
|  | |

|  |  |  |
| --- | --- | --- |
| RF-Tickets-04 | Inserter, Ediar contactos | |
| Objetivos Asociados: | Se generará un algoritmo que le permita a nuestra aplicación crear una lista de contactos y posteriormente insertará los nuevos contactos que encuentre en la BD si y solo si no están repetidos. | |
|
| Requisitos Asociados: | RF-Tickets-01 | |
|
| Descripción: | Insertar nuevos usuarios, o en caso de que el administrador requiera modificar un usuario existente, actualizarlo. | |
|
| Pre-Condición: |  | |
|
| Secuencia: | Paso: | Acción: |
| 1 | Llenar los campos del formulario de usuarios. |
| 2 | De existir el usuario, validar si quiere actualizar los campos diferentes. |
| 3 | De no existir el contacto, insertara todos los valores nuevos. |
| Post-Condición: |  | |
|
| Excepciones: |  | |
|  | |

**1.9 Requerimientos No Funcionales**

|  |  |  |
| --- | --- | --- |
| RNF-Tickets-01 | Envió de correo de confirmación. | |
| Objetivos Asociados: |  | |
|
| Requisitos Asociados: |  | |
|
| Descripción: | Enviar correo de confirmación al usuario relacionado a su ticket. | |
|
| Pre-Condición: |  | |
|
| Secuencia: | Paso: | Acción: |
| 1 | Una vez creado un ticket enviar correo de confirmación al usuario |
| 2 | De existir el ticket, enviar correo al usuario informando el nuevo status. |
| 3 |  |
| Post-Condición: |  | |
|
| Excepciones: |  | |
|  | |

|  |  |  |
| --- | --- | --- |
| RNF-Tickets-02 | Listar tickets. | |
| Objetivos Asociados: |  | |
|
| Requisitos Asociados: |  | |
|
| Descripción: | Mostrar en una lista vertical los tickets existentes. | |
|
| Pre-Condición: | Que exista por lo menos un ticket registrado. | |
|
| Secuencia: | Paso: | Acción: |
| 1 | Desplegar los ticket existentes en la BD |
| 2 |  |
| 3 |  |
| Post-Condición: |  | |
|
| Excepciones: |  | |
|  | |

|  |  |  |
| --- | --- | --- |
| RNF-Tickets-03 | Listar usuarios. | |
| Objetivos Asociados: |  | |
|
| Requisitos Asociados: |  | |
|
| Descripción: | Mostrar en una lista vertical los usuarios existentes. | |
|
| Pre-Condición: | Que exista por lo menos un usuario registrado. | |
|
| Secuencia: | Paso: | Acción: |
| 1 | Desplegar los usuarios existentes en la BD |
| 2 |  |
| 3 |  |
| Post-Condición: |  | |
|
| Excepciones: |  | |
|  | |

**1.10 Modelo de desarrollo de software**

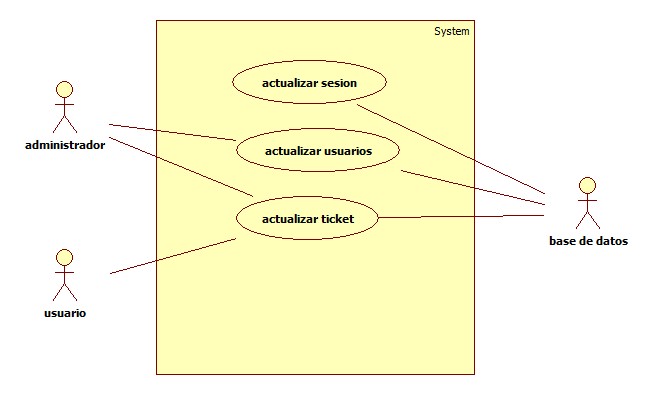
Este proyecto será realizado bajo la metodología ágil “Adaptive Software Development” por su simplicidad de implementación, y porque se basa en el desarrollo iterativo e incremental donde los requisitos y soluciones evolucionan según las necesidades del proyecto.

Se requiere además simplificar el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente.

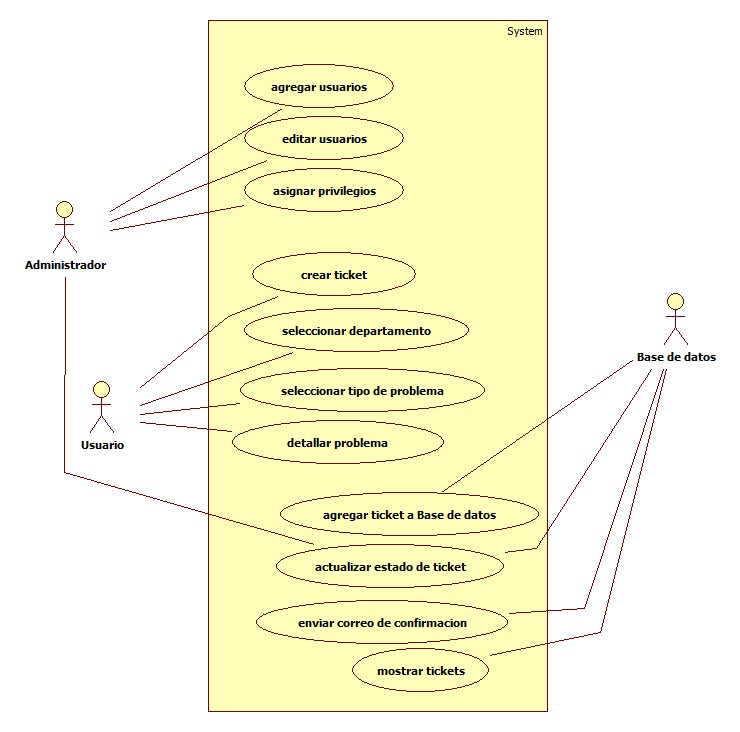
Con ayuda de Git-Hub y esta metodología será creada esta aplicación web.

**Capitulo ll Análisis**

**2.1 Diagrama de casos de uso general.**

****

**2.2 Diagrama de casos de uso detallado.**

****

**2.2.1 Escenarios de casos de uso detallado.**

|  |  |  |  |
| --- | --- | --- | --- |
| ESCENARIO DE CASO DE USO | | | |
| Número de control: | 1 |  |  |
| Nombre: | Registro de usuarios. |  |  |
| Quién comienza la acción: | Administrador de la aplicación. |  |  |
|  |  |  |
| Quién termina la acción: | Base de datos. |  |  |
|  |  |  |
| Precondiciones: | Que exista por lo menos un usuario registrado. | | |
|
| Que el usuario tenga privilegios de Administrador. | | |
|
| Post-condiciones: |  |  |  |
|  |  |  |
| Descripción: |  |  |  |
| El Administrador intenta escribir datos de un usuario.  La base de datos comprueba: |  |  |
| * Que todos los campos del formulario de usuario son validos. |  |  |
| Si la anterior comprobación es correcta, la base de datos verifica si el usuario existe, si el usuario existe actualiza los nuevos datos, si no existe registra un nuevo usuario. |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Excepciones: |  | | |
|  | | |
| Otras especificaciones: |  | | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ESCENARIO DE CASO DE USO | | | |
| Número de control: | 2 |  |  |
| Nombre: | Creación de ticket. |  |  |
| Quién comienza la acción: | Usuario. |  |  |
|  |  |  |
| Quién termina la acción: | Base de datos. |  |  |
|  |  |  |
| Precondiciones: | Que el usuario este registrado en el sistema. | | |
|
|  | | |
|
| Post-condiciones: |  |  |  |
|  |  |  |
| Descripción: |  |  |  |
| El Usuario intenta crear un ticket.  La base de datos comprueba que:     * Todos los campos del formulario de ticket esten completes. |  |  |
| Si la anterior comprobación son correctas crea y le asigna una prioridad al ticket y envia un correo de confirmación al usuario. |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Excepciones: |  | | |
|  | | |
| Otras especificaciones: |  | | |
|  | | |