Programadores anónimos

Franz Alvarado Vargas, Alberth Vivas Chavarria

Escuela de Ingeniería en Sistemas de Computación, Universidad Fidélitas, San José, Costa Rica

falvarado70869@ufide.ac.cr

avivas70762@ufide.ac.cr

***Abstract*— Este proyecto aborda la creación y mejora sobre una plataforma, debido al crecimiento tecnológico y descontrol por parte de las empresas se ha implementado la creación de una aplicación que brinde a los usuarios confiabilidad y calidad en los sistemas al momento de los reportes de fallos en los sistemas.**

Keywords- Seguridad, Calidad, problemas, evaluación.

I. Valores y acuerdos de equipo

Colaboración y comunicación promover un ambiente donde todos compartas sus ideas dudas sin temor. Una responsabilidad compartida cada miembro del equipo debe hacerse cargo de la calidad del código poder revisar el código de otros miembros y aceptar comentarios que estos puedan brindar.

Revisiones de código toda funcionalidad debe de ser revisada por otro miembro del equipo el cual pueda brindar aportes para mejora. Respeto y empatía escuchar opiniones distintas sin descalificarlas, reconocer los errores y éxitos en equipo.

II. Definición de listo (Definition of ready)

1.Componentes: descripción clara de los componentes que integran la plataforma que los usuarios puedan comprender, usar, enviar sus casos de una manera simple y clara así poder usar todos los componentes de la plataforma.

2. Datos de pruebas y validaciones: descripción de los casos de prueba validando el correcto funcionamiento, los miembros del equipo deben comprender y aprobar las asignaciones.

III. Definición de terminado (Definition of done)

1.Funcionalidad implementada: el sistema cumple con los requisitos esperados realiza correctamente las operaciones cuando un usuario logra ingresar y enviar sus reportes.

2.Documentación y código: se documenta el funcionamiento de la plataforma, el código es verificado por los miembros del equipo así mismo es aprobado y cumple con los estándares de calidad ejecutándose en el entorno previsto.

IV. Historias de usuario

a. **Ingreso de solicitud**:

¿Quién? Como cliente de la plataforma.
¿Qué?

Registrar una solicitud para brindar soporte técnico, indicando las características nombre, correo y el tipo de problema con una descripción.

¿Para qué?

Con que fin se ingresa la solitud para que soporte técnico atienda mi solicitud.

Criterios de aceptación

Requerimientos funcionales.

1. El sistema debe permitir el registro de nuevas solicitudes con los campos obligatorios.
2. Enviar una confirmación de solicitud recibida.

Requerimientos no funcionales.

1. La comunicación y seguridad debe mantenerse lo más seguro posible.
2. La plataforma debe permitir y guardar información sin perdida.

b. Visualización de solicitudes:

¿Quién?

Técnico de soporte

¿Qué?

Registrar una solicitud de soporte pendiente en la plataforma.

¿Para qué?

Priorizar y atendes los casos de los clientes de la manera más rápida

Requerimientos funcionales.

1.El sistema debe mostrar en lista las solicitudes que se encuentran sin resolver.

2.Las solicitudes ingresadas deben mostrar la información básica para poder resolver.

Requerimientos no funcionales.

1. La comunicación y seguridad debe mantenerse mediante autenticación.
2. La plataforma debe de ser rápida para mayor facilidad de los usuarios.

c. Cierre de las solicitudes

¿Quién?

Usuario

¿Qué?

Calificar la atención brindada y poder cerrar el caso de

soporte.

¿Para qué?

Indicar el nivel de atención y cerrar el caso.

Requerimientos funcionales.

1. El estado del caso debe cambiar cuando se resuelva.
2. El sistema debe indicar la información de cierre.

Requerimientos no funcionales.

1. Debe de contar una interfaz accesible para los usuarios.
2. Los cierres de los casos deben realizarse de manera rápida y ágil.

V. Diagrama de Entidad Relación (ER)

El diagrama muestra las relaciones entre las entidades en el que se puede observar relaciones como de muchos a pocos representado por la doble raya en las líneas de conexión que aunque se observa doble en ambos extremos este tipo de relación demuestra relación de muchos a pocos. Por ejemplo, un mismo usuario puede crear muchos tiquetes, pero cada tiquete pertenece a un solo usuario. El símbolo que aparece como una figura representa los usuarios a modo de ejemplo y para entenderlo mejor, la entidad Notificación muestra que un usuario con acceso por ejemplo técnico al tener un tiquete asignado puede recibir varias notificaciones para el caso pero cada notificación tiene relación con un usuario que es el que está vinculado con dicho caso.

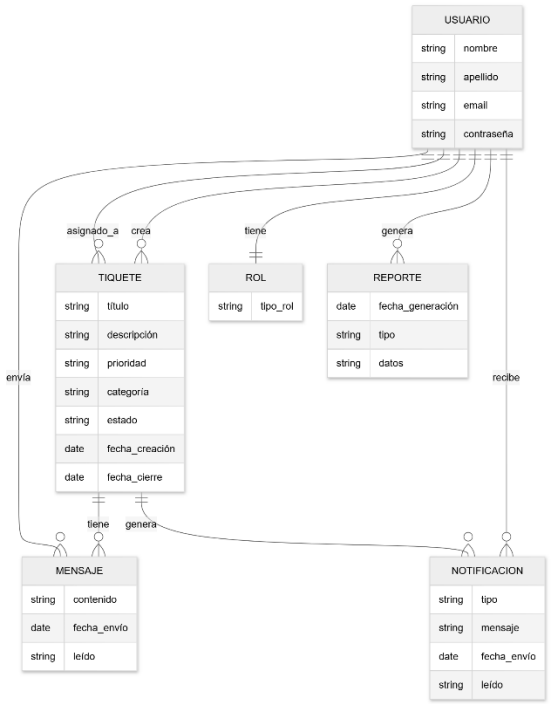

Fig. 1 Diagrama de Entidad Relación.

REFERENCIAS

[1] Atlassian. (s. f.). What is the Definition of Done (DoD) in Agile? | Atlassian. https://www.atlassian.com/agile/project-management/definition-of-done