



UNIVERSIDAD DEL ISTMO, CAMPUS IXTEPEC

LICENCIATURA EN INFORMÁTICA

ALUMNO: JOSÉ IVÁN GARCÍA GÓMEZ

PROFESOR: FLORENTINO RUIZ AQUINO

ASIGNATURA: SISTEMA DE INFORMACIÓN I

EVALUACIÓN DE CALIDAD DEL SOFTWARE APLICANDO EL
MODELO MCCALL: SISTEMA DE COMUNICACIÓN ACADÉMICA

SEMESTRE: 2025-20256 A

GRUPO: 908

FECHA: 08 DE DICIEMBRE DEL 2025

ÍNDICE

1. INTRODUCCIÓN.....	3
2. DESCRIPCIÓN DEL SISTEMA A EVALUAR	4
3. MODELO MCCAL.....	4
3.1. CORRECCIÓN.....	6
3.2. FIABILIDAD	6
3.3. EFICIENCIA	6
3.4. INTEGRIDAD	7
3.5. USABILIDAD	7
3.6. MANTENIBILIDAD	8
3.7. FLEXIBILIDAD	8
3.8. PORTABILIDAD.....	8
3.9. INTEROPERABILIDAD.....	9
4. TABLA DE EVALUACIÓN	9
5. CONCLUSIÓN.....	11
6. REFERENCIAS	11

1. INTRODUCCIÓN

El propósito de esta evaluación es calificar la calidad de una aplicación destinada a fortalecer la comunicación formal dentro de la Universidad del Istmo. Siendo construida para emplear tecnologías actuales como Expo y React Native, e integra funciones tales como el inicio de sesión mediante correo institucional y contraseña, registro de cuentas nuevas, y el envío y recepción de mensajes formales.

La evaluación tiene como objetivo determinar su grado de aceptación frente a los requerimientos institucionales, identificando fortalezas, limitaciones y elementos que pueden optimizarse para un uso productivo dentro del entorno universitario.

Se toma como referencia el Modelo de Calidad de McCall, que busca realizar una descomposición del concepto genérico de calidad en 3 capacidades o atributos: Operación, Transición y Revisión. Cada una de estas capacidades tiene a su vez un conjunto de factores que finalmente definen ciertos criterios que permiten evaluar el producto por medio de métricas, que dan cuenta de la medida en la que el sistema evaluado posee cierta característica. Estas métricas cuentan con sus propios criterios o medidas que posibilitan la medición de la calidad.



Figura 1. Atributos del modelo McCall.

2. DESCRIPCIÓN DEL SISTEMA A EVALUAR

El sistema es una aplicación desarrollada como parte de un sistema académico institucional orientado a mejorar la comunicación formal entre estudiantes, profesores y áreas administrativas de la universidad. Se centra en ofrecer un canal oficial, privado y ordenado, donde toda la interacción queda registrada dentro del entorno institucional.

El sistema está construido bajo un enfoque moderno, empleando Expo + React Native para la interfaz y herramientas externas para la gestión de usuarios y almacenamiento de mensajes. Su diseño sigue un modelo orientado a componentes, priorizando la usabilidad, la organización de información y la privacidad académica.

3. MODELO MCCAL

El sistema presenta 4 requisitos funcionales:

1. Inicio de sesión

Permite el acceso al sistema por medio de las credenciales de correo electrónico institucional y contraseña creadas por el usuario.

La imagen muestra un formulario de inicio de sesión con el título "Iniciar Sesión" en negrita. Hay dos campos de texto: "Correo institucional" y "Contraseña". Abajo de los campos hay un botón azul con el texto "Entrar". Debajo del botón, en un color más oscuro, se dice "¿No tienes cuenta? [Regístrate aquí](#)".

Figura 2. Apartado de Login.

2. Registro de Usuarios

Permite la creación de un nuevo usuario por medio de los datos de nombre completo, matrícula del estudiante o profesor, correo electrónico institucional y contraseña.



Crear cuenta

Nombre completo

Matrícula

Correo electrónico

Contraseña

Registrarme

Figura 3. Apartado de creación de cuenta.

3. Selección a quien enviarle mensajes

Al iniciar sesión se puede observar con que usuarios puedes interactuar.

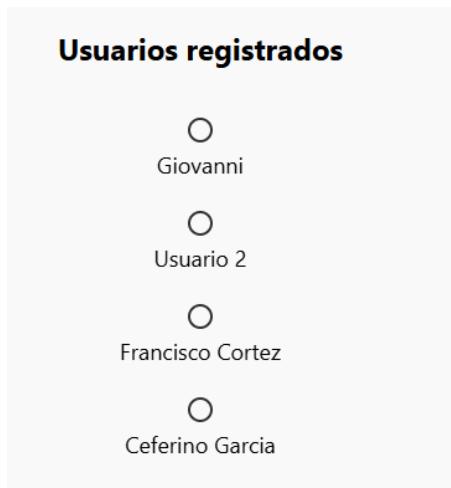


Figura 4. Apartado de usuarios registrados.

4. Envío de mensajes y muestreo en tiempo real

Se puede realizar el envío de mensajes con los usuarios registrados.

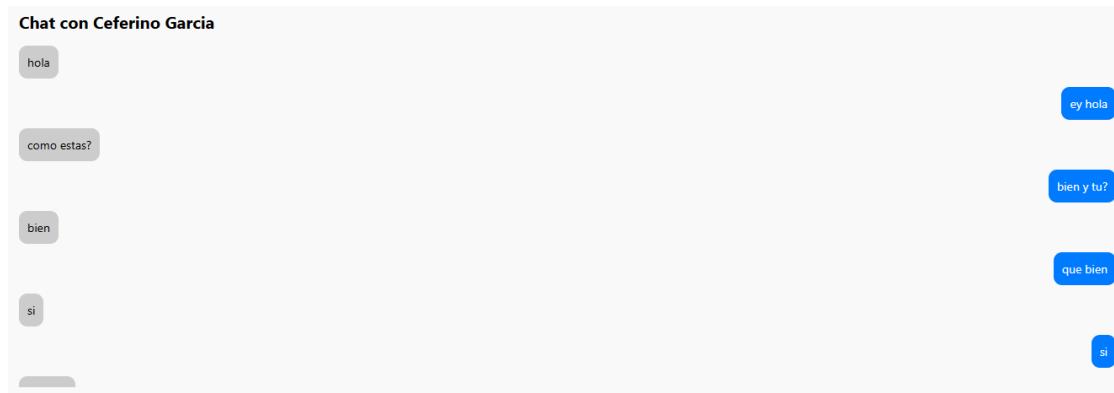


Figura 5. Chat en tiempo real.

4.1. CORRECCIÓN

Densidad en Defectos

$$D = \frac{\text{Total de errores reportados}}{\text{KLOC (Miles de líneas de código)}}$$

El sistema consta de 5 archivos TypeScript, con un promedio de 170 líneas de código por archivos. Con una estimación aproximada de 850 líneas, equivalente a 0.85 KLOC.

Se identificaron algunos problemas:

Falta de seguridad ante las contraseñas de inicio de sesión

$$D = \frac{1}{0.85 \text{ KLOC}} = 1.18$$

La densidad de defectos obtenida elevada debido al reducido tamaño del código, por lo que al ser un inicio de proyecto con un código principal reducido y un solo error, no se está cumpliendo la formula pero no necesariamente quiere decir que su funcionamiento es malo.

Cobertura de requerimientos

$$C = \frac{\text{Requerimientos probados exitosamente}}{\text{Total de requerimientos funcionales}} \times 100$$

$$C = \frac{4}{4} \times 100 = 100$$

El sistema alcanzó una cobertura del 100%, lo que indica que todos los requerimientos funcionales fueron cumplidos de manera correcta durante las pruebas realizadas, demostrando que todas las funciones evaluadas cumplen lo prometido

4.2. FIABILIDAD

$$\text{Uptime} = \frac{\text{Tiempo operativo total}}{\text{Tiempo total planificado}} \times 100$$

$$\text{Uptime} = \frac{1}{3} \times 100 = 33.33 \%$$

Nos un resultado del 33.33%, siendo correcto porque se tuvo una operación durante 1 hora de un total de 3 horas planificadas. Aunque el sistema no presentó fallas durante el periodo activo, este porcentaje muestra que aún es necesario ampliar el tiempo de pruebas y validar su comportamiento continuo.

4.3.EFICIENCIA

Se midió el consumo de memoria de JavaScript utilizando las herramientas de rendimiento del navegador. El motor de JavaScript reportó un heap total de 25.9 MB, lo cual es un valor considerablemente bajo para una aplicación móvil basada en Expo y React Native. Esto indica una buena gestión de componentes y ausencia de fugas de memoria visibles.

Además, la tasa de asignación promedio de 12.6 kB/s se mantuvo estable, sin incrementos que pudieran sugerir procesos ineficientes, renderizados innecesarios o cargas excesivas en pantalla. Estos valores se encuentran muy por debajo de los límites críticos habituales de 100 - 150 MB de JS Heap para aplicaciones React complejas, lo que indica un uso eficiente de recursos.

4.4. INTEGRIDAD

$$\text{Efectividad en el control de acceso} = \frac{\text{Intentos de accesos no autorizados}}{\text{Total de intentos de intrusión}} \times 100$$

$$\text{Efectividad en el control de acceso} = \frac{0}{10} \times 100 = 0$$

El sistema obtiene 0% de efectividad en el control de acceso, debido a que no existe un límite de intentos fallidos en el inicio de sesión. Durante las pruebas, se pudieron realizar intentos ilimitados con contraseñas incorrectas sin que el sistema aplicara bloqueos, temporizadores o alertas. Por ello, aunque la autenticación funciona, no evita accesos no autorizados, lo que reduce completamente su capacidad de protección.

4.5. USABILIDAD

Métrica 1: Tiempo de Aprendizaje

El usuario logró familiarizarse con las funciones principales, inicio de sesión, registro y uso del chat, en un tiempo promedio de 10 minutos, muy por debajo del estándar < 30 min. Esto indica que la interfaz es clara y las opciones están organizadas de manera intuitiva.

Métrica 2: Tasa de Error de Usuario

Durante las pruebas no se registraron errores al diseño de la interfaz. Las acciones realizadas al envío y recepción de mensajes, cambios entre pantallas y creación de cuentas fueron completadas sin equivocaciones.

4.6. MANTENIBILIDAD

El sistema presenta una estructura organizada, con clases bien separadas y funciones claramente definidas, lo que facilita localizar cualquier problema y aplicar correcciones sin mayor complejidad. Durante la revisión no se identificaron fallos malos, y se analizó que modificaciones como ajustar una función, cambiar elementos de la interfaz o añadir nuevas características pueden realizarse rápidamente gracias a la claridad del código. El tiempo promedio para resolver un fallo o implementar una mejora es muy poco. El MTTR estimado se mantiene entre 30 minutos y 1 hora por cambio menor, el sistema es fácil de mantener, extender y depurar sin riesgo de generar errores secundarios.

4.7. FLEXIBILIDAD

El sistema muestra una excelente forma de adaptabilidad, ya que su estructura y organización permiten agregar nuevos requerimientos sin generar complicaciones en el resto de los apartados. La separación clara entre componentes facilita modificar reglas de negocio, ajustar comportamientos específicos o añadir nuevas funciones, siempre y cuando estos cambios se mantengan alineados con el propósito original de la aplicación y no introduzcan riesgos innecesarios.

4.8. PORTABILIDAD

$$\frac{\text{Navegadores y SO soportados correctamente}}{\text{Total de entornos objetivo}} \times 100$$

Se probó únicamente en el navegador web mediante localhost

$$\frac{1}{1} \times 100 = 100 \%$$

El sistema funciona sin fallos en el navegador local, pero no se ha experimentado su comportamiento en otros entornos como dispositivos móviles, otros navegadores, etc.

4.9. INTEROPERABILIDAD

$$\frac{\text{Solicitudes de intercambio de datos existosas}}{\text{Total de solicitudes}} \times 100$$

$$\frac{10}{10} \times 100 = 100\%$$

Durante las pruebas, el chat logró comunicarse entre dos sesiones distintas una en el navegador normal y otra en modo incógnito, ambas ejecutándose desde localhost. Esto quiere decir que el sistema sí realiza intercambio de datos en tiempo real a través del backend configurado.

Para la prueba se enviaron 10 mensajes entre ambas sesiones, y los 10 fueron recibidos correctamente sin pérdidas ni retrasos significativos.

5. TABLA DE EVALUACIÓN

Instrucciones: Asigne una calificación de 1 (Pobre) al 5 (Excelente) para cada factor del sistema evaluado.

Factor McCall	Puntuación (1-5)	Observaciones / Evidencias Encontrada
Corrección	4	El sistema funciona correctamente con lo que se está presentado, inicio de sesión, registro de usuarios, chat. Funciona completamente bien, solamente con la observación de la seguridad de contraseñas.
Fiabilidad	5	No presenta ningún fallo, responde correctamente a lo que se pide, es fluido, rápido, en el chat no es lento ni siquiera falla.
Eficiencia	5	Brinda un buen rendimiento y velocidad y es aceptable al tener pocas funciones, pero al crecer se

		debe de mantener con una eficiencia.
Integridad	3	No se integra seguridad antes posibles ataques de obtención de contraseñas al tratar de adivinarlas y por no colocar un límite de intentos de contraseñas.
Usabilidad	5	Muy interactivo y fácil de usar, no es necesario buscar soluciones al momento de querer realizar la función de chat.
Mantenibilidad	5	El sistema puede ser muy fácil de mejorar, mantiene una excelente estructura, puede pasar por diferentes cambios y verse aún más profesional.
Flexibilidad	5	Es fácil de agregarle nuevos requerimientos, pero siempre y cuando no se pierda el objetivo del sistema ni sea riesgoso.
Portabilidad	4	El sistema trae una facilidad de verse en otros entornos, pero únicamente se ha probado desde un, se requiere seguir experimentando en diferentes entornos.
Interoperabilidad	5	El sistema puede ser implementando con facilidad para ser conectado con otros sistemas, por su excelente programación.
PROMEDIO FINAL	4.5	4.5/5

6. CONCLUSIÓN

La evaluación del sistema de comunicación académica demuestra que la aplicación cumple de manera sólida con los requisitos funcionales y presenta altos niveles de eficiencia, usabilidad, fiabilidad e interoperabilidad. Aunque requiere mejorar aspectos de seguridad e incrementar las pruebas en distintos entornos, el sistema muestra una arquitectura flexible, mantenible y con gran potencial para crecer y consolidarse como una herramienta institucional efectiva.

7. REFERENCIAS

M, C. (s/f). *Modelos de Evaluación de Recursos Educativos Digitales—Modelos de calidad McCall*. Recuperado el 8 de diciembre de 2025, de
<https://sites.google.com/view/evaluacionredgrupo2/modelos-de-calidad-mccall>