Tecnológico de Monterrey

Planeación de sistemas de software (Gpo 103)

Documentación: Plan de calidad

Alan De Loa Larios
Gilberto Ángel Camacho Lara
Eugenio Elizondo Jaime
Gianni André Freire Vásquez
Aldo Leonardo López Ruiz
Carlos Fernando Ramos Mena

Equipo y Roles

INTEGRANTES	ROL	RESPONSABILIDADES
Alan De Loa Larios	Ingeniero de base datos y arquitecto de software	Definir la estructura interna de los datos, así como diseñar una arquitectura adecuada para el desarrollo del proyecto y la comunicación entre sus componentes
Gilberto Ángel Camacho	Desarrollador Front End	Diseño de componentes de UI con enfoque en UX.
Eugenio Elizondo Jaime	Scrum Master y Product Owner	Dirigir y llevar el control del proyecto con metodología SCRUM, además de funcionar como un vínculo entre el cliente y los desarrolladores
Gianni André Freire	QA Engineer & tester	Asegurar la calidad del software mediante el desarrollo de pruebas unitarias y el testing de la app
Aldo Leonardo López Ruiz	Desarrollador Front End	Diseño de componentes de UI con enfoque en UX.
Carlos Fernando Ramos Mena	Desarrollador Back End	Desarrollo de endpoints para la obtención, modificación, creación y eliminación de datos. Manejo de datos seguro

Prácticas de Calidad

Para garantizar la calidad del software durante todo el ciclo de desarrollo, se implementarán diversas prácticas enfocadas en la prevención, detección y corrección temprana de defectos.

Entre las principales prácticas de calidad se incluyen:

- **Revisión de código:** Se llevarán a cabo revisiones periódicas de código por parte del equipo de desarrollo para identificar posibles errores, asegurar la adherencia a los estándares de codificación y fomentar buenas prácticas de programación.
- **Uso de estándares de desarrollo:** Se establecerán y seguirán lineamientos específicos para la escritura de código, asegurando la mantenibilidad, escalabilidad y legibilidad del software.
- **Gestión de configuración y control de versiones:** Se utilizará un sistema de control de versiones (Github) para llevar un seguimiento detallado de los cambios en el código, facilitando la colaboración y recuperación de versiones anteriores en caso de ser necesario.

Estrategia de Pruebas

Para validar el correcto funcionamiento del software y detectar errores antes de su implementación en producción, se adoptará una estrategia de pruebas estructurada, basada en la automatización y en la ejecución sistemática de diferentes tipos de pruebas.

Entre las pruebas consideradas dentro de la estrategia se incluyen:

- **Pruebas unitarias:** Se desarrollarán pruebas automatizadas para evaluar individualmente cada componente del sistema, asegurando que funcionen de manera correcta e independiente. Estas pruebas ayudarán a detectar fallos en el código antes de su integración con otros módulos.
- **Pruebas de integración:** Se validará la comunicación y el comportamiento conjunto de diferentes módulos del sistema, asegurando que la interacción entre componentes funcione correctamente.
- **Pruebas funcionales:** Se verificarán los requisitos del software mediante la ejecución de pruebas sobre los casos de uso principales, garantizando que el sistema cumpla con la funcionalidad esperada.
- **Pruebas de regresión:** Se ejecutarán pruebas automatizadas y manuales para asegurar que nuevas modificaciones en el código no afecten funcionalidades previamente implementadas.

Entregables de Calidad

Para garantizar la trazabilidad y la documentación adecuada del proceso de pruebas, se generarán diversos entregables de calidad, incluyendo:

- **Casos de prueba:** Documentación detallada de los escenarios de prueba, especificando condiciones iniciales, datos de entrada, pasos a seguir, resultados esperados y criterios de aceptación.
- **Plan de pruebas:** Documento que describe el enfoque, alcance, herramientas utilizadas y cronograma de ejecución de las pruebas, asegurando una planificación estructurada del proceso.
- **Evidencias de ejecución:** Reportes detallados de los resultados obtenidos en las pruebas, incluyendo capturas de pantalla, logs y métricas de desempeño.
- **Reportes de defectos:** Registro detallado de los errores detectados, su nivel de severidad, pasos para su reproducción y estado de resolución. Estos reportes permitirán priorizar correcciones y dar seguimiento a los problemas detectados.
- **Informe de cobertura de pruebas:** Indicador de la proporción del código que ha sido evaluado mediante pruebas automatizadas, permitiendo identificar áreas del sistema que requieren mayor validación..

Tipos de Severidades

Las severidades se llegaron a clasificar según su impacto en la plataforma, siendo estas divididas en 4 secciones:

- **Crítica:** El sistema no permite el acceso o procesamiento de información clave, como la asignación de empleados o la autenticación de usuarios.
    - Ejemplo: La IA no genera recomendaciones o la plataforma no permite iniciar sesión.
- **Alta:** Problemas graves que afectan funcionalidades clave, pero con soluciones temporales.
    - Ejemplo: Los currículums generados no contienen información completa o los datos de los empleados no se guardan correctamente.
- **Media:** Errores que afectan la experiencia, pero no impiden el uso del sistema.
    - Ejemplo: La asignación de personal tarda más de lo esperado, o hay fallos en la interfaz que requieren recargar la página.
- **Baja:** Defectos menores que no afectan la funcionalidad principal.

    - Ejemplo: Un botón con mala alineación en la UI o un mensaje de error mal escrito.

Tipos de Defectos

Los defectos en la plataforma se pueden llegar a clasificar en 5 secciones distintas:

- **Funcionales:** Errores en el flujo de asignación de empleados, generación de CVs incorrectos o fallos en las recomendaciones de IA.
- **De rendimiento:** La plataforma tarda más de 3 segundos en responder (violando el criterio de rendimiento definido).
- **De seguridad:** Brechas en la gestión de roles, permitiendo que empleados accedan a información restringida.
- **De compatibilidad:** La plataforma no se visualiza bien en ciertos navegadores o dispositivos.
- **De usabilidad:** El usuario no entiende cómo actualizar su perfil o no encuentra fácilmente las métricas de talento.

Casos de prueba

Los casos de prueba se dividen en los 4 siguientes:

Pruebas Funcionales

Caso de Prueba 1: Autenticación de Usuario

- **Objetivo:** Validar que los empleados puedan iniciar sesión correctamente.
- **Tipo:** Funcional
- **Criterios de éxito:**
    - Un usuario con credenciales correctas puede acceder a la plataforma.
    - Un usuario con credenciales incorrectas recibe un mensaje de error adecuado.
    - El sistema bloquea la cuenta tras 5 intentos fallidos.

Caso de Prueba 2: Generación de Currículums

- **Objetivo:** Comprobar que el sistema genere CVs con la información completa.
- **Tipo:** Funcional
- **Criterios de éxito:**
    - El CV incluye nombre, experiencia, habilidades y formación académica.
    - La descarga del CV en PDF se realiza correctamente.
    - No hay datos truncados o ausentes.

Caso de Prueba 3: Asignación de Empleados a Proyectos

- **Objetivo:** Verificar que el sistema asigne empleados según compatibilidad.
- **Tipo:** Funcional
- **Criterios de éxito:**
    - El sistema sugiere empleados con habilidades y experiencia alineadas con el proyecto.
    - La asignación se refleja en la base de datos correctamente.
    - Un manager no puede asignar empleados sin los permisos adecuados.

Caso de Prueba 4: Sugerencias de Cursos según Brechas de Habilidad

- **Objetivo:** Verificar que el sistema recomienda cursos adecuados a los empleados con base en sus habilidades y metas.
- **Tipo:** Funcional
- **Criterios de éxito:**
    - El sistema identifica correctamente las brechas de habilidades del empleado.
    - Los cursos sugeridos están alineados con la carrera y objetivos del usuario.
    - Se muestra una explicación clara de por qué se recomienda cada curso.
    - Si el usuario ya ha completado un curso, este no se vuelve a sugerir.

Pruebas No Funcionales

Caso de Prueba 5: Rendimiento de la Plataforma

- **Objetivo:** Medir el tiempo de respuesta del sistema.
- **Tipo:** No funcional (Pruebas de rendimiento)
- **Criterios de éxito:**
    - Las consultas a la base de datos tardan menos de 1 segundo.
    - La carga de la interfaz no supera los 3 segundos.
    - Con 100 usuarios simultáneos, la plataforma sigue operando sin caídas.

Caso de Prueba 6: Seguridad y Cifrado de Datos

- **Objetivo:** Verificar que los datos de los empleados estén protegidos.
- **Tipo:** No funcional (Pruebas de seguridad)
- **Criterios de éxito:**
    - La información sensible (salarios, evaluaciones) está cifrada en la base de datos.
    - Un empleado no puede acceder a los datos de otro usuario sin los permisos adecuados.
    - Se registra cada acceso a información confidencial para auditoría.

Caso de Prueba 7: Experiencia de Usuario

- **Objetivo:** Evaluar la facilidad de uso del sistema.
- **Tipo:** No funcional (Pruebas de usabilidad)
- **Criterios de éxito:**
    - Los usuarios pueden utilizar la aplicación correctamente después de un entrenamiento de 2 horas
    - Los managers encuentran rápidamente la información clave en el dashboard.
    - El diseño es responsive y funciona en computadoras de escritorio y laptops

Pruebas de Integración

Caso de Prueba 8: Integración entre IA y el Sistema de Asignaciones

- **Objetivo:** Asegurar que la IA genera recomendaciones correctas y las envía al módulo de asignación.
- **Tipo:** Integración

- **Criterios de éxito:**
    - La IA recibe correctamente los datos de habilidades y experiencia del empleado.
    - Las recomendaciones de la IA se reflejan en el sistema de asignaciones.
    - No hay pérdidas de datos en la transferencia entre módulos.

Caso de Prueba 9: Integración con el Sistema de Correo Electrónico

- **Objetivo:** Verificar que el sistema envíe notificaciones correctamente.
- **Tipo:** Integración
- **Criterios de éxito:**
    - Los empleados reciben correos cuando hay cambios en su asignación.
    - Los managers reciben alertas cuando hay empleados disponibles.
    - No hay correos duplicados ni errores en el envío.

Pruebas de Regresión

Caso de Prueba 10: Validación de Funcionalidades tras Actualizaciones

- **Objetivo:** Garantizar que tras un cambio en el código, las funciones existentes sigan operando correctamente.
- **Tipo:** Regresión
- **Criterios de éxito:**
    - Después de una actualización en la generación de CVs, los CVs anteriores siguen generando correctamente.
    - La autenticación y asignación de empleados siguen funcionando sin errores.
    - No aparecen nuevas fallas en módulos previamente estables.

Métricas de Calidad

Para evaluar y mejorar continuamente la calidad del software, se medirán los siguientes indicadores clave que permitirán analizar el desempeño de los procesos de desarrollo y prueba.

- **Code Coverage:** Porcentaje del código fuente que ha sido ejecutado por las pruebas automatizadas.
- **Defect Rate (Tasa de defectos):** Relación entre el número de defectos encontrados y el total de casos de prueba ejecutados.
- **Defectos por cambio de software:** Número de errores introducidos por cada modificación en el código.
- **Escaped Bugs:** Cantidad de defectos que no fueron detectados en las pruebas y llegaron a producción.
- **Tiempo medio de resolución de defectos:** Tiempo promedio que se tarda en identificar, analizar y corregir un defecto desde su detección hasta su cierre.