**Plan de pruebas**

**Plataforma de Administración Estudiantil - Colegio Santa Margarita de Escocia.**

***Fecha:* 12/08/2024**

**Tabla de contenido**

**Información del proyecto**

**Datos**

| Integrantes | Leandro Plaza Jimmy Rojas Matias Cid Benjamin Vidal |
| --- | --- |
| Representante de la organización |  |
| Nombre del proyecto | **Plataforma de administración estudiantil- Colegio Santa Margarita de Escocia.** |
| Fecha de preparación | 12/08/2024 |
| Período del proyecto | 23/11/2024 |

**Historial de versiones**

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| 11-11-2024 | 1.0 | QA Senior | Equipo de desarrollo (QA) | Estructura base del plan de pruebas |
| 11-11-2024 | 2.0 | QA Junior | Equipo de desarrollo (QA) | Delimitar pruebas a realizar |
| 11-11-2024 | 3.0 | QA Senior | Equipo de desarrollo (QA) | Corroborar posibles fallos |

**Aprobaciones**

| **Nombre y Apellido** | **Cargo** | **Departamento u organización** | **Firma** |
| --- | --- | --- | --- |
| Benjamin Vidal | Lider de proyecto | Equipo de desarrollo (Gestión) |  |
| Matias Cid | Product Owner | Equipo de desarrollo (Gestión) |  |

**Resumen Ejecutivo**

Este documento ha sido elaborado para definir técnicas, herramientas y actividades clave en la ejecución y validación del plan de pruebas. Incluye la asignación de responsabilidades para cada tarea, los recursos necesarios, los alcances y los prerrequisitos que deben considerarse en el desarrollo de cada prueba. Su objetivo es asegurar el cumplimiento de los requisitos establecidos.

**Alcance de las pruebas**

**Elementos de pruebas**

Módulo - Gestión de horarios :

* Prueba de creación de horarios sin conflictos: verificar que no se permite el solapamiento entre clases.
* Modificación de horarios existentes: probar la actualización en tiempo real para todos los usuarios afectados.
* Consulta de horario según rol (estudiante, profesor, apoderado): asegurar que cada rol acceda solo a su horario personalizado.

Módulo - Mensajería Interna

* Envío de comunicados: prueba que tanto el profesor como el administrador puedan enviar comunicados a los grupos correctos (estudiantes, apoderados o ambos).
* Envío de mensajes entre usuarios: verificar que estudiantes y apoderados puedan enviar mensajes a los profesores.
* Recepción y organización de mensajes: asegurar que los mensajes se ordenen por fecha y sean accesibles en un historial.

Módulo - Publicación de material didáctico

* Publicación de material: probar que el profesor pueda subir distintos formatos de archivo y asignar destinatarios.
* Recepción y descarga de material: asegurar que los estudiantes puedan ver y descargar los materiales de manera correcta.
* Publicación y respuesta de tareas: verificar que los estudiantes puedan ver, descargar y responder a las tareas.

Módulo - Notificaciones Automáticas

* Envío de notificaciones por correo: probar que se envíen notificaciones para tareas y materiales nuevos.
* Recordatorio de pagos y entrevistas: verificar la entrega de recordatorios en tiempo y con la información correcta.
* Notificaciones internas: asegurarse de que las notificaciones de nuevos mensajes lleguen a los usuarios en la plataforma según se requiera.

Módulo - Agendamiento de Entrevistas

* Agendamiento y confirmación de entrevistas: verificar que los apoderados puedan reservar y que los profesores puedan confirmar o rechazar con motivo.
* Gestión de disponibilidad: probar que los profesores puedan establecer sus horarios de manera precisa.
* Registro de acta de entrevista: validar que el profesor pueda registrar los detalles de la reunión, con respaldo visible.

Restricciones de Usuarios

* Prueba de acceso según rol: verificar que cada rol (estudiante, profesor, administrador, apoderado) solo pueda acceder a las funcionalidades correspondientes. Ejemplo: un apoderado no debe acceder a la creación de horarios.
* Control de permisos: asegurar que acciones críticas (como modificación de horarios, envío de comunicados o publicación de materiales) estén restringidas a usuarios autorizados únicamente.
* Bloqueo por intentos fallidos: probar que se limite el acceso tras varios intentos de inicio de sesión fallidos.
* Gestión de roles dinámicos: validar que cualquier cambio de rol de usuario refleje los accesos y restricciones en tiempo real.

Pruebas Responsivas

* Pruebas en dispositivos móviles: verificar que la plataforma sea completamente funcional en diferentes tamaños de pantalla (teléfonos, tablets y computadoras).
* Prueba de orientación: asegurar que la interfaz se ajuste bien en modos horizontal y vertical.
* Pruebas de escalado de componentes: validar que los elementos de la UI se reorganicen adecuadamente y no se superpongan o desborden en diferentes resoluciones y pantallas.
* Pruebas de funcionalidad táctil: para dispositivos móviles, verificar que los botones, menús y opciones sean fácilmente accesibles y funcionales.

**Criterios de aceptación**

**Criterios de aceptación o rechazo**

Completar con el 100% de las pruebas unitarias e integración, al menos el 95% de las pruebas funcionales y el 90% de las pruebas no funcionales. De cuyo total de casos de prueba debe llegar al 92% de los casos exitosos. Se espera que la cobertura de los defectos corregidos responda a la plenitud de sus incisos, así como la cobertura de todos los componentes y líneas de código modificadas por las mismas correcciones.

**Criterios de suspensión**

Bajo las implicancias de defectos o fallos de un módulo que imposibiliten la cadena de interacciones del usuario cliente frente a procedimientos secundarios al de compra, se considera como suspendida a la principal prueba responsable de dicho suceso, de modo que pueda ser enfocada como un error modular y que se pueda reintegrar correctamente a posteriori en la interfaz de usuario. Se determina un porcentaje del 5% como máximo de casos fallidos o similares para su aceptación.

**Criterios de reanudación**

En caso de ser pruebas suspendidas de gran implicancia en el principal funcionamiento del aplicativo; la venta de productos, o de ser casos suspendidos que gracias a su ausencia afecten notoriamente la integridad de la plataforma misma, se espera la reanudación del caso de prueba. Considerando un máximo de casos suspendidos como 5%, en caso de superar el valor antes mencionado, se establece que deben ser reanudados los casos necesarios para delimitar dicho porcentaje al necesario para la aceptación del caso de prueba.

**Entregables**

**Recursos**

**Requerimientos de entornos – Hardware**

Servidor de Pruebas (o Máquina Virtual)

* CPU: 4 núcleos mínimo (Intel i5 o equivalente en AMD para entorno local).
* RAM: 8 GB mínimo
* Almacenamiento: SSD de 256 GB mínimo.
* Conectividad: Red de alta velocidad

Equipo Local para QA y Desarrolladores

* CPU: 4 núcleos mínimo (Intel i5 o equivalente).
* RAM: 8 GB mínimo.
* Almacenamiento: SSD de 256 GB mínimo.
* Resolución de Pantalla: 1920x1080

Dispositivos Móviles

* Mínimo un dispositivo con iOS (iPhone/iPad) y otro con Android.
* Opcional: Emuladores de dispositivos móviles.

**Requerimientos de entornos – Software**

Sistema Operativo

* Windows 10/11, Linux (Ubuntu 20.04+), o macOS 10.15+.

Lenguajes y Entornos de Desarrollo

* Python 3.13.0
* Django versión 5.1.2
* MySQL versión 9.1.0.

Bases de Datos y Almacenamiento

* Instancia de MySQL similar al entorno de producción, con datos de prueba relevantes.
* Opcional: Docker para crear entornos aislados para bases de datos y servidores de aplicaciones.

Navegadores (para pruebas de interfaz)

* Google Chrome (última versión).
* Mozilla Firefox (última versión).
* Safari (última versión).

**Personal**

El equipo encargado de las pruebas de este proyecto está conformado por roles clave que garantizan una cobertura completa y efectiva de las actividades de validación y verificación. El Product Owner participa en la definición de los criterios de aceptación y asegura que las pruebas estén alineadas con los objetivos del negocio. El Scrum Master supervisa la planificación y ejecución de las pruebas, asegurando que el equipo siga las mejores prácticas y que los plazos sean cumplidos. Los QA Senior y QA Junior lideran las pruebas funcionales, de integración, y de sistema, diseñando casos de prueba exhaustivos y ejecutando escenarios que validen tanto el comportamiento como la usabilidad del sistema.

Por otro lado, el Fullstack Developer Senior colabora en la creación de pruebas unitarias y en la automatización de procesos críticos, asegurando la estabilidad de los módulos desarrollados, mientras que el Fullstack Developer Junior da soporte en la ejecución de pruebas técnicas y corrección de errores detectados, aprendiendo y contribuyendo activamente al fortalecimiento del producto. Este equipo, con roles bien definidos, garantiza que el sistema cumpla con los estándares de calidad y las expectativas de los usuarios finales.

**Planificación y organización**

**Procedimientos para las pruebas**

Basado principalmente en pruebas dinámicas cuya ejecución del software será la principal metodología a utilizar con el fin de medir el nivel de calidad con la que este fue codificado y el nivel de cumplimiento en relación con la especificación del sistema. Como nuestros lineamientos de desarrollo se basan en metodologías ágiles, es necesario delimitar la exclusividad de ejecución de las actividades mencionadas con anterioridad por sobre el resto, para reducir al mínimo el total de actividades relacionadas con el proyecto y optimizar los tiempos de ejecución.

La ejecución de pruebas debe iniciar con la creación de los datos de prueba necesarios para ejecutar los casos de prueba diseñados, su ejecución será principalmente automatizada por lo que se espera un estudio y desarrollo previo por parte del equipo de pruebas. En caso de detección de fallo en el sistema, este debe ser documentado y registrado dentro del framework de Protractor. Una vez el defecto ha sido corregido, en caso de ser viable su corrección, es necesario realizar un nuevo testeo que permita asegurar una exitosa solución. Finalmente, es fundamental ejecutar un ciclo de regresión que nos permita garantizar, que los defectos corregidos en el proceso de pruebas general, no haya generado otros tipos de defectos en el sistema.

**Matriz de responsabilidades**

| Actividad | Product Owner | Scrum Master | QA Senior | QA Junior | Fullstack Dev Senior | Fullstack Dev Junior |
| --- | --- | --- | --- | --- | --- | --- |
| Definición de estrategia de pruebas | C | S | R | A | C | C |
| Desarrollo de casos de prueba | A | - | R | C | C | A |
| Ejecución de pruebas unitarias | - | S | - | - | R | R |
| Ejecución de pruebas funcionales | - | S | R | C | C | - |
| Ejecución de pruebas de integración | - | S | R | C | A | A |
| Ejecución de pruebas de sistema | - | S | R | C | A | A |
| Ejecución de pruebas de aceptación | R | - | A | A | A | - |
| Automatización de pruebas | - | S | R | A | A | A |
| Documentación de resultados | - | S | R | R | A | A |
| Análisis de defectos | A | - | R | A | C | C |
| Revisión de corrección de errores | R | - | R | A | R | R |
| Validación final antes de despliegue | R | S | R | - | A | A |
| CLAVE: R = Responsabilidad Principal; S = Supervisión; A = Apoyo; C = Consultado. | | | | | | |

**Tabla de Requerimientos**

| Requerimiento 1 | Gestión de Horarios |
| --- | --- |
| Requerimiento 2 | Mensajería Interna |
| Requerimiento 3 | Publicación de Material Didáctico |
| Requerimiento 4 | Notificaciones Automáticas |
| Requerimiento 5 | Agendamiento de Entrevistas |

**Casos de prueba**

Basándonos en los requerimientos obtenidos para esta segunda etapa de desarrollo del sistema se procede con la creación de los casos de prueba y su posterior realización. Es por ello que se crea el documento: “*10. Casos de prueba.xlsx”* donde se detallan cada una de las pruebas a realizar y los resultados obtenidos de la misma.