

1. Documentación de Pruebas

1.1 Casos de Prueba

Descripción

Se presenta a continuación la matriz de casos de prueba diseñada para evaluar las funcionalidades del software, garantizando la calidad y el cumplimiento de los requisitos.

ID C as o	Módulo	Descripción	Entrada Esperada	Resultado Esperado	Fecha	Esta do
C P0 01	Autentica ción	Login con credenciales correctas	Usuario y contraseña válidos	Acceso permitido	2024-12- 05	Apr oba do
C P0 02	Autentica ción	Login con credenciales incorrectas	Usuario y contraseña inválidos	Acceso denegado	2024-12- 10	Apr oba do
C P0 03	Código Diario	Generación automática de código	Solicitud diaria automática	Código aleatorio generado	2024-12- 15	Apr oba do
C P0 04	Registro de Jornada	Inicio de jornada con código válido	Código correcto, agente identificado	Inicio de cronómetro	2024-12- 20	Apr oba do
C P0 05	Registro de Jornada	Inicio de jornada con código inválido	Código incorrecto, agente identificado	Error en inicio de cronómetro	2025-01- 05	Apr oba do
C P0 06	Consulta de Jornada	Consultar horas laboradas por agente	Solicitud de consulta	Reporte con horas trabajadas	2025-01- 10	Apr oba do
C P0 07	Correcci ón de Registro	Modificación de horas por administrador	Solicitud de corrección	Registro actualizado	2025-01- 15	Apr oba do
C P0 08	Segurida d	Bloqueo de cuenta tras múltiples intentos fallidos	Usuario con múltiples intentos fallidos	Cuenta bloqueada	2025-01- 20	Apr oba do

C P0 09	Notificaciones	Envío de notificación de código diario	Generación de código	Notificación enviada correctamente	2025-01-25	Aprobado
---------------	----------------	--	----------------------	------------------------------------	------------	----------

1.2 Pruebas Unitarias

Objetivo:

Verificar la correcta ejecución de funciones individuales dentro del backend y frontend.

Herramientas:

- JUnit para pruebas de backend en Spring Boot.
- Jasmine para pruebas unitarias de funciones JS.
- Postman para pruebas de controladores REST.

Casos evaluados:

- Métodos de autenticación.
- Generación de códigos aleatorios.
- Validación de códigos.
- Inicio y detención de cronómetro.
- Validación de datos de entrada.

1.3 Pruebas de Integración

Objetivo:

Garantizar que los módulos se comuniquen correctamente entre sí.

Herramientas:

- Postman para pruebas de API REST.
- Hibernate Validator para validaciones de datos.
- H2 Database para simulación de la base de datos en memoria.

Casos evaluados:

- Comunicación entre el frontend y backend por medio de AJAX.
- Persistencia de datos en la base de datos MySQL.

- Flujo completo desde la autenticación hasta el registro de horas.
- Manejo de transacciones.
- Validación de integridad referencial.

1.4 Pruebas de Sistema

Objetivo:

Validar que el sistema completo cumpla con los requisitos funcionales y no funcionales.

Casos evaluados:

- Flujo de inicio de sesión y registro de jornada.
- Consulta de reportes por agente.
- Envío de código desde administrador a tienda.
- Modificación de registros por administrador.
- Seguridad en el manejo de sesiones.
- Rendimiento del sistema con carga de múltiples usuarios.
- Desconexión automática por inactividad.

1.5 Reporte de Errores y Correcciones Aplicadas

ID Error	Módulo	Descripción	Fecha	Estado	Corrección Aplicada
ER001	Autenticación	Error al validar credenciales con usuario inactivo	2024-12-07	Corregido	Se añadió validación de estado de usuario
ER002	Código Diario	Código duplicado generado en la misma fecha	2024-12-16	Corregido	Algoritmo de generación modificado para evitar duplicados
ER003	Registro de Jornada	Cronómetro no se detiene al finalizar jornada	2024-12-22	Corregido	Se ajustó lógica de detención del cronómetro
ER004	Consulta de Jornada	Error al mostrar reportes con múltiples agentes	2025-01-12	Corregido	Query SQL optimizado

ER 00 5	Segurida d	No se bloqueaba la cuenta tras múltiples intentos fallidos	202 5- 01- 18	Co rre gid o	Se implementó lógica de bloqueo de cuenta
---------------	---------------	--	------------------------	-----------------------	--

1.6 Conclusión

Se realizaron pruebas unitarias, de integración y de sistema, logrando asegurar que el software cumpla con los requisitos establecidos. Los errores identificados fueron corregidos y se verificó su correcta solución. Además, se implementaron medidas de seguridad adicionales y se garantizó la eficiencia del sistema bajo carga moderada.