**Plan de Pruebas**

***Caso: Bot Construye+***

***Fecha: [18/10/2024]***

**Asignatura: Portafolio de Título**

**Integrantes:**

**Diego Orellana**

**David Fernandez**

**Gabriel Florit**

**José Barboza**

**Fecha de entrega: 18/10/2024**

**Profesor: Arturo Vargas**

**Historial de revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 18/10/2024 | <1.0> | Se desarrolla el documento incluyendo todos los puntos referentes a: Introducción, Requerimiento para pruebas, Estrategia de pruebas, Recursos, Actividades, Hitos y Entregables | Gabriel Florit  Davis Fernandez |

**TABLA DE CONTENIDO**

[**1.**](#_heading=h.gjdgxs) **Introducción** 4

[**1.2**](#_heading=h.30j0zll) **Alcance de las pruebas** 4

[**1.2.1**](#_heading=h.1fob9te) **Elementos de pruebas** 4

[**1.2.2**](#_heading=h.3znysh7) **Pruebas Funcionales** 4

[**1.2.3**](#_heading=h.2et92p0) **Riesgos** 5

[**1.2.3.1**](#_heading=h.tyjcwt) **Matrices de riesgos** 5

[**1.2.3.1.1**](#_heading=h.3dy6vkm) **Pruebas** 5

[**1.3**](#_heading=h.1t3h5sf) **Tipos de pruebas** 5

[**1.3.1**](#_heading=h.4d34og8) **Pruebas Funcionales** 5

[**1.3.2**](#_heading=h.2s8eyo1) **Herramientas involucradas** 5

[**1.3.3**](#_heading=h.17dp8vu) **Preparación del ambiente de pruebas** 6

[**1.3.4**](#_heading=h.3rdcrjn) **Diseño del ambiente de pruebas** 6

[**1.3.5**](#_heading=h.26in1rg) **Diseño ambiente de pruebas** 9

[**2.**](#_heading=h.lnxbz9) **Resultados de las pruebas** 9

[**2.1 Reporte de defectos** 10](#_heading=h.44sinio)

1. **Introducción**
   1. **Resumen ejecutivo**

El proyecto Construye++ se centra en desarrollar un chatbot que facilite el acceso a beneficios y servicios para los trabajadores de empresas miembros de la Cámara Chilena de la Construcción (CChC). Dada su importancia social, el éxito del proyecto depende de garantizar que las funcionalidades del sistema operen de manera eficiente y segura, brindando una experiencia de usuario satisfactoria. El presente documento detalla el Plan de Pruebas diseñado para evaluar el correcto funcionamiento del chatbot, cubriendo una amplia gama de casos de uso esenciales para asegurar su estabilidad, rendimiento y seguridad en un entorno real.

Objetivos del Plan de Pruebas

* Validar la funcionalidad: Verificar que cada una de las características principales del chatbot funcione como se espera.
* Asegurar la seguridad: Garantizar que las medidas de seguridad, como autenticación y protección de datos, se implementen de manera efectiva.
* Probar la usabilidad: Evaluar la experiencia del usuario y la accesibilidad del sistema en múltiples dispositivos y navegadores.
* Monitorear el rendimiento: Comprobar que el sistema mantenga un rendimiento óptimo, incluso con múltiples usuarios concurrentes.
* Garantizar la calidad: Verificar que la interfaz de usuario sea intuitiva y visualmente coherente, y que no existan errores de ortografía o gramática en las respuestas del chatbot.

En los próximos pasos, desarrollaremos una matriz de pruebas específica para cada uno de los casos de uso detallados.

* 1. **Alcance de las pruebas**
     1. **Elementos de pruebas**

El **plan de pruebas** para el proyecto **Construye++** abarca todas las funcionalidades del chatbot, garantizando que cumpla con los requisitos definidos y opere de manera efectiva en diferentes escenarios. Se pondrán a prueba los principales casos de uso que los usuarios y administradores experimentarán en su interacción con la plataforma. Además, se verificarán la integración con los sistemas de la **CChC** y los aspectos de seguridad, rendimiento y usabilidad en diversos dispositivos y navegadores.

**Áreas cubiertas**:

* Registro y autenticación de usuarios.
* Consultas y acceso a beneficios.
* Actualización de información personal.
* Gestión de certificados y documentos.
* Notificaciones, asistencia técnica y personalización de alertas.
* Integración con sistemas de terceros y seguridad de datos.
* Rendimiento bajo diferentes niveles de carga.

Los **elementos de pruebas** incluyen los siguientes componentes del sistema:

* **Chatbot**: Comportamiento del chatbot, procesamiento de consultas y generación de respuestas.
* **Interfaces de usuario**: Todas las interfaces de usuario, tanto en dispositivos móviles como en navegadores, serán evaluadas.
* **Backend empresarial de la CChC**: Validación de la correcta integración con los sistemas internos de la CChC, incluyendo acceso a bases de datos de beneficios, eventos y recursos.
* **Módulo de notificaciones**: Verificación del envío, recepción y personalización de notificaciones a los usuarios.
* **Sistema de autenticación**: Pruebas de seguridad, autenticación de usuarios, y posibles medidas adicionales como la autenticación de dos factores (2FA).
* **Dashboard administrativo**: Funcionalidades del dashboard para la gestión de usuarios, reportes, estadísticas y configuración del sistema.
  + 1. **Pruebas Funcionales**

Las **pruebas funcionales** verificarán que el sistema cumpla con las funcionalidades establecidas en los requisitos del proyecto. Algunas de las pruebas clave incluyen:

1. **Registro y autenticación**:
   * Probar el registro en el chatbot ingresando datos como nombre, RUT, nombre y RUT de la empresa CChC.
   * Validar el inicio de sesión utilizando credenciales correctas y incorrectas, verificando el sistema de autenticación y posibles bloqueos por múltiples intentos fallidos.
2. **Consulta de beneficios**:
   * Probar la consulta de beneficios específicos para la membresía de cada usuario, incluyendo acceso a descuentos, eventos y servicios exclusivos.
3. **Actualización de datos personales**:
   * Verificar que el usuario pueda actualizar correctamente su información personal (dirección, teléfono, etc.) y que los cambios se reflejen en el backend.
4. **Solicitud de certificados**:
   * Probar la solicitud y generación de certificados de membresía, constancias de eventos y documentos relacionados con la afiliación.
5. **Recepción y personalización de notificaciones**:
   * Validar que el usuario reciba notificaciones de eventos, talleres, y recomendaciones de seguridad, y que pueda personalizar el tipo de notificaciones que desea recibir.
6. **Interacciones con el soporte técnico**:
   * Verificar que el chatbot ofrezca asistencia técnica eficiente para resolver problemas como errores de inicio de sesión o dificultades de navegación.
7. **Integración con sistemas externos**:
   * Probar la correcta integración con los sistemas de beneficios, capacitación, y asistencia social de la CChC.
8. **Rendimiento y usabilidad**:
   * Realizar pruebas de carga para asegurar que el chatbot maneje múltiples usuarios concurrentes sin degradación en el rendimiento.
     1. **Riesgos**

Los **riesgos potenciales** asociados al proyecto **Construye++** incluyen:

1. **Fallas en la integración con los sistemas de la CChC**:
   * Las dependencias del backend empresarial y otros sistemas externos podrían generar fallos o retrasos en el acceso a datos, afectando la experiencia del usuario.
2. **Riesgos de seguridad**:
   * Vulnerabilidades en la autenticación o en el manejo de datos sensibles podrían comprometer la privacidad y seguridad de los usuarios. Es esencial garantizar medidas robustas de protección de datos y cifrado.
3. **Problemas de rendimiento bajo carga alta**:
   * El chatbot podría experimentar una disminución del rendimiento o tiempos de respuesta más lentos si hay un gran número de usuarios simultáneos, afectando la experiencia del usuario.
4. **Compatibilidad de dispositivos y navegadores**:
   * Inconsistencias en el comportamiento de la plataforma en diferentes dispositivos o navegadores podrían afectar la usabilidad y accesibilidad para algunos usuarios.
5. **Errores en la personalización y envío de notificaciones**:
   * Fallos en el sistema de notificaciones podrían resultar en información desactualizada o incorrecta enviada a los usuarios, generando confusión o desconfianza.
6. **Riesgos asociados a la escalabilidad**:
   * Si el sistema no está adecuadamente preparado para escalar, podría tener problemas para soportar un crecimiento rápido en el número de usuarios y datos manejados.

Este **plan de pruebas** detallará cada uno de estos aspectos para minimizar los riesgos y asegurar la entrega de un producto robusto y confiable.

* + - 1. **Matrices de riesgos**
         1. **Pruebas**

| **N° Riesgo** | **Descripción** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1   |  | | --- |  | Fallas en la integración con sistemas externos | | --- | | Fallas en la integración con sistemas externos | Alta | Realizar pruebas de integración tempranas y establecer canales de comunicación con proveedores. |
| 2 Vulnerabilidades de seguridad | Fallos en la autenticación o cifrado que puedan comprometer datos sensibles. | Alta | Implementar autenticación robusta (incluyendo 2FA) y asegurar el cifrado de datos en tránsito. |
| 3 Rendimiento bajo carga alta | El sistema podría degradarse con múltiples usuarios simultáneos. | Media | Optimizar código, realizar pruebas de carga y establecer alertas para monitorear el rendimiento. |
| 4 Incompatibilidad con dispositivos y navegadores | Inconsistencias en la interfaz o funcionalidades en diferentes plataformas y navegadores. | Media | Probar exhaustivamente en dispositivos y navegadores más utilizados, y ajustar la compatibilidad. |
| 5 Fallos en el sistema de notificaciones | Envío incorrecto de notificaciones o fallos en la personalización de alertas. | Media | Verificar el sistema de notificaciones, realizar pruebas funcionales y validar en producción. |
| 6 Pérdida de datos en la actualización de información personal | Posibilidad de que los datos actualizados no se guarden correctamente. | Media | Implementar validaciones y mecanismos de confirmación en la actualización de información. |
| 7 Problemas de escalabilidad | El sistema podría no soportar un crecimiento rápido de usuarios o datos. | Alta | Realizar pruebas de escalabilidad y establecer arquitectura flexible para el crecimiento. |
| 8 Errores en el dashboard administrativo | Fallos en las funcionalidades de administración, como la gestión de usuarios o generación de reportes. | Media | Realizar pruebas exhaustivas del dashboard, especialmente en el manejo de roles y permisos. |
| 9 Error en las respuestas del chatbot | Respuestas inexactas o incompletas por parte del chatbot. | Media | Validar el procesamiento del lenguaje natural y realizar pruebas con escenarios de conversación. |
| 10   |  | | --- |  | Insuficiencia en medidas de redundancia y monitoreo | | --- | | Fallos en el sistema sin detección o recuperación oportuna. | Alta | Implementar mecanismos de redundancia y un sistema de monitoreo continuo para prevenir caídas. |

* 1. **Tipos de pruebas**
     1. **Pruebas Funcionales**

Estas pruebas aseguran que el sistema cumpla con los requisitos funcionales especificados para el proyecto. Dado que el chatbot Construye++ está completamente en la nube (Azure), es importante validar todas las interacciones y funcionalidades que dependen del backend alojado en Azure.

##### **1.1 Registro de Usuario en el Chatbot**

* **Objetivo de la prueba:**Verificar el correcto registro de usuarios, guardando datos como nombre, RUT, empresa y RUT de la empresa en la base de datos.
* **Técnicas a utilizar:**
  + Testing de caja negra.
  + Pruebas de límite.
* **Criterios de validación:**
  + Registro exitoso con datos válidos.
  + Mensajes de error claros para datos inválidos.
  + Almacenamiento correcto en la base de datos.

##### 

##### 

##### 

##### **1.2 Iniciar Sesión**

* **Objetivo de la prueba:**Validar el correcto inicio de sesión con usuario y contraseña.
* **Técnicas a utilizar:**
  + Pruebas de autenticación.
  + Pruebas de seguridad (bloqueos tras múltiples intentos fallidos).
* **Criterios de validación:**
  + Inicio de sesión exitoso con credenciales válidas.
  + Rechazo de credenciales incorrectas.
  + Registro de intentos fallidos y bloqueo.

##### **1.3 Consulta de Beneficios**

* **Objetivo de la prueba:**Verificar la correcta respuesta del sistema a consultas sobre beneficios específicos.
* **Técnicas a utilizar:**
  + Pruebas basadas en escenarios.
  + Pruebas de regresión.
* **Criterios de validación:**
  + Respuesta correcta y clara a las consultas.
  + Enlaces e información adicional relevante proporcionada.

##### **1.4 Actualización de Información Personal**

* **Objetivo de la prueba:**Asegurar la actualización correcta de datos personales (dirección, teléfono, etc.).
* **Técnicas a utilizar:**
  + Pruebas de modificación de datos.
  + Validación de formatos de campos.
* **Criterios de validación:**
  + Datos personales actualizados correctamente.
  + Rechazo de entradas inválidas.

##### 

##### 

##### 

##### 

##### **1.5 Solicitud de Certificados**

* **Objetivo de la prueba:**Verificar la generación y entrega de certificados de membresía o constancias solicitadas por los usuarios.
* **Técnicas a utilizar:**
  + Testing de generación de documentos.
  + Pruebas de rendimiento.
* **Criterios de validación:**
  + Generación correcta del certificado solicitado.
  + Confirmación y almacenamiento de las solicitudes.

##### **1.6 Recepción de Notificaciones**

* **Objetivo de la prueba:**Validar el envío de notificaciones personalizadas y generales (eventos, recomendaciones, capacitaciones).
* **Técnicas a utilizar:**
  + Pruebas de notificaciones automáticas.
  + Pruebas de personalización de preferencias.
* **Criterios de validación:**
  + Envío correcto y oportuno de notificaciones.
  + Personalización de notificaciones según preferencias del usuarios
    1. **Herramientas involucradas**

* Postman: Para pruebas de API y simulación de solicitudes HTTP.
* Azure Monitor: Para revisar logs de actividad, errores y supervisión del rendimiento.
* SQL Server Management Studio (SSMS): Para verificar la correcta inserción y actualización de datos en la base de datos Azure SQL.
* Azure Active Directory: Para pruebas de autenticación de usuarios.
* Azure Application Insights: Para revisar eventos y errores en el sistema.
* Azure Logic Apps: Para validar y gestionar flujos de información entre el chatbot y el backend.
* Bot Framework Emulator: Para pruebas en un entorno simulado del chatbot.
* Azure Blob Storage: Para el almacenamiento de certificados y otros documentos generados.
* Azure Notification Hubs: Para gestionar el envío de notificaciones push a los usuarios.
  + 1. **Preparación del ambiente de pruebas**

Para garantizar un entorno de pruebas adecuado, el ambiente de pruebas debe replicar las condiciones del entorno de producción, especialmente considerando que Construye++ se ejecuta completamente en la nube mediante Azure. A continuación, se detallan los pasos para preparar este ambiente:

#### **Pasos para la preparación del ambiente de pruebas:**

1. **Configuración del Entorno en Azure:**
   * Crear un entorno de prueba aislado en Azure para evitar interferencias con el entorno de producción.
   * Clonar las bases de datos de producción (sin datos sensibles) en Azure SQL Database para garantizar integridad y consistencia en las pruebas.
   * Habilitar entornos de prueba para las siguientes aplicaciones y servicios:
     + Azure Web App para la instancia del chatbot.
     + Azure App Services para las APIs que manejan las consultas y actualizaciones de usuarios.
     + Azure Notification Hubs para el envío de notificaciones de prueba.
2. **Implementación de Seguridad:**
   * Configurar Azure Active Directory para manejar las credenciales de los usuarios de prueba y su autenticación.
   * Implementar autenticación de dos factores (2FA) en el ambiente de pruebas para replicar las políticas de seguridad del entorno productivo.
3. **Configuración de Monitoreo y Logging:**
   * Activar **Azure Monitor** y **Application Insights** para capturar datos de rendimiento y posibles errores.
   * Establecer umbrales de alerta para tiempos de respuesta lentos o caídas en servicios críticos.
4. **Datos de Prueba:**
   * Crear un conjunto de datos de prueba realistas que incluyan usuarios, empresas, beneficios y certificados para que las pruebas sean representativas.
   * Asegurarse de que los datos utilizados en el ambiente de pruebas no contengan información confidencial.
5. **Integración Continua y Despliegue:**
   * Configurar pipelines de integración continua y entrega continua (CI/CD) en **Azure DevOps** para automatizar el despliegue del chatbot y sus actualizaciones en el ambiente de pruebas.
     1. **Diseño del ambiente de pruebas**

El diseño del ambiente de pruebas debe emular el entorno productivo de **Construye++**, dado que el sistema es enteramente basado en la nube. A continuación, se presenta la estructura y los componentes clave de este ambiente.

1. **Infraestructura Cloud en Azure:**
   * **Azure App Services**: Aloja el backend del chatbot y las APIs, simulando el ambiente productivo.
   * **Azure SQL Database**: Base de datos para almacenar la información de usuarios, beneficios, certificados, entre otros.
   * **Azure Logic Apps**: Orquesta las integraciones con otros sistemas (Recursos Humanos, Atención Médica, Gestión Financiera).
   * **Azure Notification Hubs**: Envío de notificaciones push a los dispositivos móviles de los usuarios.
   * **Azure Blob Storage**: Almacena documentos generados, como certificados, para garantizar su accesibilidad.
   * **Azure Active Directory (AAD)**: Gestión de usuarios y autenticación segura.
2. **Ambiente de Simulación de Chatbot:**
   * **Bot Framework Emulator**: Simulación de conversaciones en tiempo real para pruebas manuales y automáticas del chatbot.
3. **Monitoreo y Logging:**
   * **Azure Monitor**: Controla el estado y rendimiento de las aplicaciones en tiempo real.
   * **Azure Application Insights**: Recoge métricas detalladas sobre el rendimiento y los errores.

### **Ventajas del Ambiente de Pruebas**

* **Escalabilidad en la Nube**: Al estar basado en Azure, el entorno de pruebas puede escalarse fácilmente para simular diferentes niveles de carga de usuarios y tráfico de consultas.
* **Entorno Aislado**: El ambiente está completamente separado del entorno de producción, lo que garantiza que los errores de prueba no afecten las operaciones diarias del sistema en producción.
* **Monitoreo Completo**: Herramientas como Azure Monitor y Application Insights proporcionan visibilidad en tiempo real sobre el rendimiento y los problemas durante las pruebas, lo que facilita la identificación y corrección de errores rápidamente.
* **Repetibilidad**: Con el uso de pipelines de CI/CD, el despliegue de nuevas versiones o correcciones del chatbot se puede realizar de forma automatizada, asegurando que el entorno de pruebas siempre esté actualizado.
* **Seguridad**: Al implementar autenticación y protección de datos en el entorno de pruebas, se replican las mismas medidas de seguridad que en el ambiente productivo, permitiendo una evaluación exhaustiva de la seguridad.

### **Desventajas del Ambiente de Pruebas**

* **Costos Adicionales**: La creación de un entorno de pruebas en la nube con los servicios de Azure puede generar costos adicionales, dependiendo del uso de recursos como bases de datos, almacenamiento y servicios de notificación.
* **Latencia**: Aunque se puede simular el rendimiento del entorno productivo, las pruebas en un entorno de pruebas en la nube pueden sufrir latencias distintas, lo que podría no reflejar completamente el comportamiento en producción.
* **Complejidad**: El diseño y la configuración de un entorno de pruebas completo en la nube requiere una planificación cuidadosa y conocimiento profundo de los servicios de Azure, lo que podría aumentar la complejidad y tiempo de preparación.
* **Datos No Realistas**: Los datos utilizados en el entorno de pruebas, aunque simulen condiciones reales, no siempre reflejan la complejidad y variedad de datos que podrían encontrarse en producción.

Este diseño de ambiente de pruebas proporciona un balance entre la simulación precisa de las condiciones de producción y la flexibilidad que ofrece la nube, permitiendo pruebas exhaustivas antes del despliegue final.

1. **Resultados de las pruebas**

Para el resultado de las pruebas, se han seleccionado y documentado 11 casos de uso a modo de ejemplo. Estos casos representan una muestra significativa del funcionamiento esperado del sistema, pero no constituyen la totalidad de las pruebas planificadas, ya que se evaluarán todos los casos de uso identificados una vez que se ejecute el plan completo.

**Caso de Uso 1: Registrarme en el chatbot proporcionando mi nombre, RUT, nombre de empresa y RUT de empresa de la CChC**

* **Paso**: Ingresar los datos de registro (nombre, RUT, nombre de empresa y RUT de empresa).
* **Propósito**: Verificar que el chatbot registre correctamente los datos del usuario y de la empresa.
* **Condiciones**: El usuario no debe estar previamente registrado y debe proporcionar un RUT válido.
* **Resultado esperado**: El chatbot debe confirmar que el registro ha sido exitoso y almacenar la información en la base de datos.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### **Caso de Uso 2: Iniciar sesión utilizando mis credenciales (usuario y contraseña)**

* **Paso**: Ingresar el nombre de usuario y contraseña en el chatbot.
* **Propósito**: Verificar que el sistema permita el inicio de sesión solo con credenciales válidas.
* **Condiciones**: El usuario debe estar previamente registrado y las credenciales deben ser correctas.
* **Resultado esperado**: El chatbot debe permitir el acceso al usuario si las credenciales son correctas, o denegar el acceso si no lo son.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### 

### 

### 

### 

### **Caso de Uso 3: Consultar los beneficios específicos de mi membresía (descuentos, acceso a eventos, servicios exclusivos)**

* **Paso**: Solicitar los beneficios disponibles mediante comandos en el chatbot.
* **Propósito**: Verificar que el chatbot muestre los beneficios específicos según el tipo de membresía del usuario.
* **Condiciones**: El usuario debe estar autenticado y la membresía debe estar activa.
* **Resultado esperado**: El chatbot debe mostrar una lista de los beneficios aplicables al usuario.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### **Caso de Uso 4: Actualizar mi información personal (dirección, número de teléfono, etc.)**

* **Paso**: Ingresar los datos actualizados en el chatbot.
* **Propósito**: Verificar que el sistema permita la actualización de la información personal de los usuarios.
* **Condiciones**: El usuario debe estar autenticado y los nuevos datos deben ser válidos.
* **Resultado esperado**: El chatbot debe confirmar que la actualización ha sido exitosa y reflejar los nuevos datos.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### **Caso de Uso 5: Solicitar certificados de membresía o constancias de participación en eventos**

* **Paso**: Solicitar el certificado o constancia a través del chatbot.
* **Propósito**: Verificar que el chatbot permita la generación y entrega de certificados de membresía y otros documentos.
* **Condiciones**: El usuario debe estar autenticado y cumplir con los requisitos para solicitar el documento.
* **Resultado esperado**: El chatbot debe generar el documento y enviarlo al usuario en formato digital (PDF).
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### 

### **Caso de Uso 6: Recibir notificaciones y recomendaciones de seguridad**

* **Paso**: Revisar las notificaciones automáticas generadas por el chatbot.
* **Propósito**: Verificar que el chatbot envíe notificaciones periódicas sobre temas de seguridad laboral.
* **Condiciones**: El usuario debe estar suscrito a las notificaciones.
* **Resultado esperado**: El chatbot debe enviar notificaciones con información útil y relevante sobre seguridad en el trabajo.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### **Caso de Uso 7: Recibir información sobre eventos, talleres y capacitaciones organizados por la CChC**

* **Paso**: Solicitar o recibir información sobre eventos a través del chatbot.
* **Propósito**: Verificar que el chatbot envíe y muestre información actualizada sobre eventos y capacitaciones.
* **Condiciones**: El usuario debe estar autenticado y estar dentro de la lista de participantes del evento.
* **Resultado esperado**: El chatbot debe mostrar una lista de próximos eventos con detalles como fecha, ubicación y temas.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### **Caso de Uso 8: Personalizar las notificaciones que recibo (tipo de eventos)**

* **Paso**: Configurar preferencias de notificaciones en el chatbot.
* **Propósito**: Verificar que el chatbot permita personalizar el tipo de notificaciones que el usuario desea recibir.
* **Condiciones**: El usuario debe estar autenticado.
* **Resultado esperado**: El chatbot debe permitir al usuario seleccionar el tipo de notificaciones y guardarlas en su perfil.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### 

### **Caso de Uso 9: Recibir asistencia técnica para problemas relacionados con el chatbot**

* **Paso**: Solicitar asistencia técnica a través del chatbot.
* **Propósito**: Verificar que el chatbot redirija al usuario a un equipo de soporte técnico o resuelva problemas comunes.
* **Condiciones**: El usuario debe estar autenticado y haber experimentado un problema técnico.
* **Resultado esperado**: El chatbot debe proporcionar una solución o redirigir al soporte técnico.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### **Caso de Uso 10: Acceder a la documentación relevante (manuales, guías, políticas, etc.)**

* **Paso**: Solicitar la documentación a través del chatbot.
* **Propósito**: Verificar que el chatbot pueda entregar documentos relevantes de manera rápida.
* **Condiciones**: El usuario debe estar autenticado.
* **Resultado esperado**: El chatbot debe proporcionar enlaces a los documentos solicitados o entregarlos directamente.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

### **Caso de Uso 11: Realizar encuestas periódicas para evaluar la satisfacción del chatbot**

* **Paso**: Participar en una encuesta de satisfacción en el chatbot.
* **Propósito**: Verificar que el chatbot permita la creación, envío y recepción de encuestas.
* **Condiciones**: El usuario debe estar autenticado y recibir la encuesta de forma automatizada.
* **Resultado esperado**: El chatbot debe registrar las respuestas de los usuarios y generar un reporte.
* **Resultado obtenido**: **Prueba aún no realizada**.
* **Observaciones**: **Prueba pendiente**.

**2.1 Reporte de defectos**

Dado que las pruebas aún no han sido realizadas, el registro de defectos no ha sido generado. Esto se llevará a cabo una vez que se ejecute el plan de pruebas en su totalidad. Los defectos que surjan durante las pruebas se registrarán y documentarán en detalle, identificando su naturaleza, impacto y las acciones correctivas necesarias para asegurar el correcto funcionamiento del sistema y su alineación con los requisitos del proyecto.

| **ID** | **ID Caso Prueba** | **Nº Ciclo** | **Fecha** | **Módulo** | **Descripción Detallada** | **Tipo** | **Severidad** | **Estado** | ***Observación de equipo de desarrollo*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |