



CHATLY

# Chatly

## *Plan de Pruebas*

Versión <1.0.0>



# CHATLY

© 2023 Chatly  
TODOS LOS DERECHOS RESERVADOS

Queda reservado el derecho de propiedad de este documento, con la facultad de disponer de él, publicarlo, traducirlo o autorizar su traducción, así como reproducirlo total o parcialmente, por cualquier sistema o medio.

No se permite la reproducción total o parcial de este documento, ni su incorporación a un sistema informático, ni su locación, ni su transmisión en cualquier forma o por cualquier medio, sea éste escrito o electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo y escrito de los titulares de los derechos y del copyright.

FOTOCOPIAR ES DELITO.

Otros nombres de compañías y productos mencionados en este documento, pueden ser marcas comerciales o marcas registradas por sus respectivos dueños.



## Información General

TÍTULO: Chat Web Grupal - Chatly

SUBTÍTULO: Plan de Pruebas

VERSIÓN: 1.0.0

AUTOR: Grupo N° 11

## Firmas y Aprobaciones

ELABORADO POR: Grupo N°11  
*Equipo de desarrollo*

FECHA: 2024-11-30

Firma: \_\_\_\_\_

REVISADO POR: Ing. Ángel Marcelo Rea Guamán.  
*Docente*

FECHA: 2024-12-03

Firma: \_\_\_\_\_

## Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0.0	2024-11-30	Grupo N°11	Emisión Inicial



## Contenido

<b>1 Introducción</b>	<b>5</b>
1.1 Objetivos de las Pruebas	5
1.2 Alcance de las pruebas	5
1.3 Términos y Definiciones	5
<b>2 Elementos Software a Probar</b>	<b>6</b>
2.1 Especificaciones a probar	6
<b>3 Características y Tipos de pruebas</b>	<b>5</b>
3.1 Selección de tipos de pruebas	7
<b>4 Criterios de Aceptación y Rechazo</b>	<b>8</b>
<b>5 Criterios de Pruebas</b>	<b>8</b>
5.1 Criterio de Entradas	8
5.2 Criterio de Salida	8
<b>6 Entorno de pruebas</b>	<b>9</b>
6.1 Necesidades del ambiente de prueba (HW Y SW)	9
<b>7 Roles y Responsabilidades</b>	<b>9</b>
<b>8 Necesidades de formación del personal</b>	<b>10</b>
<b>9 Riesgos asumidos por el plan y planes de contingencias-restricciones</b>	<b>10</b>
<b>10 Entregables y Resultados</b>	<b>11</b>
<b>11 Compromiso del cumplimiento del plan de pruebas</b>	<b>11</b>
<b>12 Anexos</b>	<b>12</b>
12.1 Prueba Estática - SonarQube	12
12.2 Prueba Dinámica - Selenium IDE	13
12.3 Prueba de Integración	21



## Chat Web Grupal - Chatly

### 1 Introducción

#### 1.1 Objetivos de las Pruebas

El objetivo principal de este plan de pruebas es garantizar que las funcionalidades del sistema Chatly, específicamente relacionadas con el módulo de acceso y gestión de salas grupales, operen conforme a los requerimientos definidos.

Esto incluye la validación de flujos críticos como el registro de usuarios, inicio de sesión y creación de salas. Además, se busca asegurar la integración entre componentes y verificar la experiencia de usuario, priorizando la usabilidad y el rendimiento.

#### 1.2 Alcance de las pruebas

Con respecto al alcance de las pruebas tenemos: enfoque exclusivamente en las funcionalidades clave del sistema Chatly:

- Registro de Usuario: Validación de datos de entrada, manejo de errores y creación exitosa de un perfil.
- Inicio de Sesión: Verificación de autenticación mediante el ingreso del usuario y contraseña.
- Gestión de Salas: Creación de salas con nombre y descripción.
- No se incluyen pruebas de integración con redes sociales ni funcionalidades futuras no especificadas.
- Los elementos a evaluar son:
  - o Software: Módulo de Gestión de Acceso y creación de salas.
  - o Documentación: Plan de pruebas del Módulo de Gestión de Acceso y creación de salas.

#### 1.3 Términos y Definiciones

**Sala de Chat:** Espacio virtual donde los usuarios se comunican en tiempo real.

**JWT (JSON Web Token):** Estándar de autenticación para sesiones seguras.

**Plan de Pruebas:** Documento que describe la estrategia para validar un sistema.

**WebSocket:** Protocolo para comunicación bidireccional en tiempo real.

**Software:** Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

**Interfaz:** Es la parte de un sistema con la que interactúan los usuarios, es decir son los elementos visuales con los que el usuario final realiza acciones y recibe información.

**Plan de pruebas:** Es un artefacto que sirve de guía para la realización de las pruebas, y permite verificar que el sistema de información cumple las necesidades establecidas por el usuario, con las debidas garantías de calidad.

**Artefacto:** Es un producto tangible resultante del proceso de desarrollo de software, los cuales ayudan a la descripción de la función, la arquitectura o el diseño del software.



## 2 Elementos Software a Probar

### 2.1 Especificaciones a probar

Se va a probar la funcionalidad de los siguientes componentes:

- Registro de usuario mediante un formulario.
- Ingreso de sesión con usuario y contraseña.
- Validación de campos en los formularios de registro, inicio de sesión y cambio de contraseña.

A continuación, se especifica los casos de prueba funcionales a usar:

CP-1		
<b>Objetivo</b>	Verificar que estén todas las interfaces acordadas (Registro, Inicio de Sesión).	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe tener acceso a CHATLY mediante la Web.	
<b>Entradas</b>	<b>Resultados esperados</b>	<b>Resultados de las pruebas</b>
Ingresar a la aplicación	Visualización de la interfaz principal al ingresar por primera vez, interfaz de inicio de sesión con su respectivo formulario, interfaz de registro con su respectivo formulario.	OK

CP-2		
<b>Objetivo</b>	Validar los campos del formulario de: Registro de Usuario	
<b>Prioridad</b>	Media	
<b>Precondiciones</b>	El usuario debe tener acceso a CHATLY mediante la Web.	
<b>Entradas</b>	<b>Resultados esperados</b>	<b>Resultados de las pruebas</b>
Ingresar los datos en el campo: Nombre de usuario	El nombre de usuario debe tener máx 16 l min 4 caracteres. El nombre de usuario solo puede contener letras, números, guiones y puntos.	OK
Ingresar los datos en el campo: Contraseña	El campo es obligatorio y se debe ingresar una cadena min de 6 caracteres esta debe ser combinada entre mayúscula, número, caracteres especiales para así poder medir el grado de seguridad en la contraseña siendo esta (Weak, Medium, Strong).	OK
Ingresar la Confirmación de la contraseña	Se debe ingresar tal cual la contraseña antes escrita para poder registrarse, caso contrario mostrará el mensaje "Contraseña no coincide".	OK



CP-3		
<b>Objetivo</b>	Verificar la funcionalidad del registro de usuario.	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe tener acceso a CHATLY mediante la Web y conexión a internet.	
<b>Entradas</b>	<b>Resultados esperados</b>	<b>Resultados de las pruebas</b>
Clic en el botón "Registrarse"	Visualización de la interfaz de Registro. Con su respectivo formulario	OK
Llenar los campos con: Nombre de usuario, contraseña y confirmación de contraseña.	Al llenar los campos y dar clic en el botón "Registrarse", la aplicación muestra un mensaje de "Usuario registrado correctamente".	OK
Llenar los campos con datos incorrectos	Al llenar los campos y dar clic en el botón "Registrarse", la app mostrará un mensaje de "ERROR"	OK

CP-4		
<b>Objetivo</b>	Verificar la funcionalidad del inicio de sesión	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe tener acceso a CHATLY mediante la Web.	
<b>Entradas</b>	<b>Resultados esperados</b>	<b>Resultados de las pruebas</b>
Clic en el botón "Ingresar"	Visualización de la interfaz de Ingresar, con su respectivo formulario.	OK
Llenar el campos de Contraseña, con los datos correctos	Al llenar los campos y dar clic en el botón "Ingresar", la app dirige al usuario a la interfaz Home.	OK
Llenar los campos con datos incorrectos.	Al llenar los campos y dar clic en el botón "Iniciar Sesión", la app mostrará un mensaje de "ERROR, Usuario o contraseña incorrectos".	OK

CP-5		
<b>Objetivo</b>	Validar los campos de creación de una sala de chat grupal.	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe tener acceso a CHATLY mediante la Web.	
<b>Entradas</b>	<b>Resultados esperados</b>	<b>Resultados de las pruebas</b>



## CHATLY

Clic en el botón "Agregar"	El campo es obligatorio y se debe ingresar el nombre de la sala y la descripción de la misma.	OK
No llenar el campo nombre de sala.	Si no se llena el campo "nombre", aparece un mensaje de "El nombre es requerido "	OK
No llenar el campo descripción de sala.	Si no se llena el campo "descripción", aparece un mensaje de "La descripción es requerida "	OK

Con respecto a las pruebas estáticas se tiene:

- Lista de verificación para la revisión de la ERS, la cual se adjunta **G11 Lista Verificación**.
- Plataforma SonarQube para la evaluación del código fuente.

Para las pruebas dinámicas se contempla usar Selenium IDE.





### 3 Características y Tipos de pruebas

En esta sección, se presentan los tipos de pruebas que se pretenden realizar en este proyecto, las cuales reflejan las características de las pruebas a ejecutar.

#### Técnicas Estáticas:

Con respecto a la revisión de la técnica estática, que en este caso es la lista de verificación, se procede con una Revisión Guiada con los siguientes integrantes:

- Autor de los Artefactos: Alisson Clavijo y Joao Morales
- Revisor Guía: Alisson Clavijo
- Revisor: Docente

Documento/Artefacto (y versión / fecha)	Creado o Disponible		Recibido ó Revisado		Autor o Recursos	Notas
	Si	No	Si	No		
Requerimientos y Definiciones	X		X		G#11	
Cronograma completo del Proyecto	X		X		G#11	
Funciones y Reglas del Negocio	X		X		G#11	
Modelo/Flujo de Negocios	X		X		G#11	
Especificaciones de Casos de Uso	X		X		G#11	
Especificaciones en Prototipos	X		X		G#11	
Mecanismos de Arquitectura BD	X		X		G#11	
Modelo de Datos	X		X		G#11	
Lineamientos de Arquitectura	X		X		G#11	
Modelo de Producción		X		X	G#11	
Objetos de BD (Revisión de estándares)	X			X	G#11	
Objetos de Aplicativo (Revisión de estándares))	X			X	G#11	
Plan de despliegue		X		X	G#11	
Documento de Seguridades		X		X	G#11	
Matriz de Versionamiento	X		X		G#11	
Scripts		X		X	G#11	
Páginas web, entre otros	X		X		G#11	

Pruebas Estáticas		
<b>Pruebas de calidad del código</b>	Objetivo de la prueba	Analizar e identificar errores, inconsistencias y posibles mejoras en el código de la app, con el fin de corregirlos y garantizar que la calidad del código es realmente óptima y que la probabilidad de tener errores o bugs en la codificación sea la menor posible.



	Descripción de la prueba	<ul style="list-style-type: none"><li>Una vez levantado el SonarQube y configurado el archivo sonar.properties en el directorio del proyecto, se ejecuta la herramienta.</li></ul>
	Consideraciones Especiales	<ul style="list-style-type: none"><li>Se debe revisar la documentación de SonarQube para un uso correcto y asegurarse que los directorios del proyecto a evaluar sean correctos.</li></ul>

Pruebas Dinámicas		
Pruebas en Selenium IDE	Objetivo de la prueba	El objetivo de esta prueba es garantizar que el sistema permita a los usuarios iniciar sesión con credenciales válidas y que genere un mensaje de error en caso de introducir credenciales incorrectas. Además, se valida que el flujo de inicio de sesión redirija al usuario a la página principal tras una autenticación exitosa. y que la creación de una sala sea correcta.
	Descripción de la prueba	<ul style="list-style-type: none"><li>Configurar un usuario con credenciales válidas en la base de datos.</li><li>Configurar Selenium IDE para interceptar y observar las solicitudes al endpoint de inicio de sesión, registro y creación de sala.</li></ul>
	Consideraciones Especiales	<ul style="list-style-type: none"><li><b>Tiempo de Respuesta:</b> El tiempo de carga de la página tras una autenticación exitosa debe ser inferior a 3 segundos.</li></ul>



### 3.1 Selección de tipos de pruebas

En este apartado, se enlistan los tipos de pruebas a realizar, con el fin de garantizar la calidad y fiabilidad del módulo de gestión de accesos y creación de sala en CHATLY.

Características de calidad	Selección del tipo de prueba		Prueba obligatoria
	Sí	No	
PRUEBAS ESTÁTICAS			
OBJETOS DE BD (Revisión de estándares)		X	
OBJETOS DE APLICATIVO (Revisión de estándares))	X		
PRUEBAS DE CALIDAD DE CÓDIGO(Sonar)	X		
PRUEBAS DE CAJA BLANCA			
PRUEBAS UNITARIAS	X		
PRUEBAS DE CAJA NEGRA			
PRUEBAS EN Selenium IDE	X		
PRUEBAS NO FUNCIONALES			
PRUEBAS DE CARGA Y ESTRÉS	X		
PRUEBAS DE RECUPERACIÓN Y TOLERANCIA A FALLAS		X	
PRUEBAS DE SEGURIDAD Y CONTROL DE ACCESO	X		
PRUEBAS DE ESTILO	X		
PRUEBAS DE USABILIDAD	X		
PRUEBAS FUNCIONALES			
PRUEBAS DE INTEGRACIÓN	X		
PRUEBAS DE REGRESIÓN	X		
PRUEBAS DE SERVICIOS WEB	X		



PRUEBAS DE COMPATIBILIDAD Y CONVERSIÓN		X	
PRUEBAS DE INTEGRIDAD DE DATOS Y BASE DE DATOS	X		
PRUEBAS DE ACEPTACIÓN		X	
PRUEBAS DE SISTEMA	X		

## 4 Criterios de Aceptación y Rechazo

Criterios de Aceptación:

- Completar el 100% de las pruebas unitarias.
- Alcanzar al menos un 95% de casos exitosos en las pruebas funcionales.
- Garantizar una cobertura de pruebas que incluya todos los componentes y funcionalidades del módulo de gestión de acceso y creación de sala .
- Corregir el 100% de los defectos críticos y de alta prioridad identificados durante las pruebas.
- 

Criterios de Rechazo:

- No cumplir con el 100% de las pruebas unitarias.
- Tener más del 5% de casos fallidos en las pruebas funcionales.
- No cubrir todos los componentes y funcionalidades del módulo en las pruebas.
- No corregir todos los defectos críticos y de alta prioridad antes de la entrega.

## 5 Criterios de Pruebas

### 5.1 Criterio de Entradas

Las pruebas de sistema pueden comenzar cuando:

- Los elementos de hardware y software estén listos en el ambiente de Pre-Producción.
- El equipo haya terminado todo el desarrollo, realizado las pruebas unitarias de todas las características, entregado el documento de pruebas y realizado las correcciones planificadas de los defectos para la versión.
- El equipo de desarrollo ha proporcionado el software al área de calidad.
- Los casos de prueba dinámica están preparados en Selenium IDE.

### 5.2 Criterio de Salida

Las pruebas de sistema terminan cuando:

- No hay cambios en el diseño, código o características, excepto para abordar los defectos identificados durante la prueba de sistema.



- El equipo ejecuta todas las pruebas planificadas para el software.
- El equipo resolvió todos los defectos que debían ser corregidos.
- El equipo verificó que todas las cuestiones en el sistema de seguimiento de defectos estén cerradas, con pruebas de regresión.
- El equipo tendrá una reunión de salida de la fase de pruebas de sistema y estará de acuerdo en que hemos terminado la prueba de sistema.
- Se haya firmado el acta de aceptación de paso a producción.

## 6 Entorno de pruebas

### 6.1 Necesidades del ambiente de prueba (HW Y SW)

Recurso	Cantidad	Especificaciones
<b>Servidor de base de datos</b>	1	
Nombre de la Base de Datos		Postgres
Almacenamiento		10gb
<b>Dispositivo móvil</b>	1	
Sistema Operativo		Android 8.0 o superior
RAM		4 GB
Memoria		1 GB Libre
Red		WIFI

## 7 Roles y Responsabilidades

En la siguiente tabla se describen los roles y responsabilidades que debe cumplir el personal involucrado en la evaluación de la app.

Nombres y Apellidos	Rol	Responsabilidades
Docente	QA	
Alisson Clavijo Johao Morales	Equipo de proyecto	



## 8 Necesidades de formación del personal

Área	Nombres y Apellidos	Conocimientos en
Desarrollador	Alisson Clavijo	Desarrollo de backend-frontend y lenguajes de programación
Desarrollador	Johao Morales	Desarrollo de backend-frontend y lenguajes de programación
QA	Docente	Pruebas de software y herramientas de pruebas

## 9 Riesgos asumidos por el plan y planes de contingencias-restricciones

Riesgo	Estrategia de Mitigación	Contingencia (Si el riesgo se hace realidad)
Retraso en la entrega de la aplicación para pruebas.	Mejorar la comunicación del equipo y Monitorear el progreso siguiendo un cronograma.	Ejecutar las pruebas críticas y evitar las que no son esenciales.
Que no se cumplan los prerrequisitos de un caso de prueba	Se deberá verificar si los prerrequisitos se cumplen antes de poner en marcha el plan de pruebas.	Documentar el fallo sucedido detalladamente.
Falta o abandono del tester designado para las pruebas.	Capacitar a otros miembros del equipo para que realice la ejecución de las pruebas.	Designar a un tester suplente que se haga responsable de las pruebas.
Falta de acceso a entornos y datos de prueba.	Verificación previa de que los entornos de pruebas estén configurados correctamente y se tenga acceso a los respectivos datos.	Tener un entorno de prueba alternativo o datos de prueba simulados en caso de no poder acceder a los originales.



## 10 Entregables y Resultados

Una vez ejecutado y finalizado el plan de pruebas, se espera es lo siguiente:

- Plan de pruebas con sus respectivos anexos.
- Informe de resultados de las pruebas realizado por el QA
- Documentación de pruebas por el equipo de QA.

## 11 Compromiso del cumplimiento del plan de pruebas

Nombres y Apellidos	Cargo	Fecha	Firma
Docente	QA		
Alisson Clavijo	Director de Proyecto		
Johao Morales	Director de Proyecto		



## 12 Anexos

### 12.1 Prueba Estática - SonarQube

Creamos un proyecto local nuevo en Sonarqube el cual nos dará el siguiente comando para poder analizar nuestro aplicativo.

**1 Provide a token** Analyze "pruebaCRUDTareas": sqp\_52f53511ad719d3dc8d20f5407557efa291fbf36

**2 Run analysis on your project**

What option best describes your project?

**Maven** **Gradle** **.NET** **Other (for JS, TS, Go, Python, PHP, ...)**

What is your OS?

**Linux** **Windows** **macOS**

**Download and unzip the Scanner for Linux**

Visit the [official documentation of the Scanner](#) to download the latest version, and add the `bin` directory to the `PATH` environment variable

**Execute the Scanner**

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner \
  -Dsonar.projectKey=pruebaCRUDTareas \
  -Dsonar.sources=. \
  -Dsonar.host.url=https://localhost:9000 \
  -Dsonar.token=sqp_52f53511ad719d3dc8d20f5407557efa291fbf36
```

Please visit the [official documentation of the Scanner](#) for more details.

Observar los resultados en sonarqube, con los siguientes resultados:

**main** 345 Lines of Code Version **not provided** Set as homepage

**Quality Gate** Passed

**New Code** **Overall Code**

<b>Security</b> 0 Open issues 0 H 0 M 0 L	<b>Reliability</b> 0 Open issues 0 H 0 M 0 L	<b>Maintainability</b> 1 Open issues 1 H 0 M 0 L
<b>Accepted issues</b> 0 Valid issues that were not fixed	<b>Coverage</b> 0.0% On 169 lines to cover.	<b>Duplications</b> 0.0% On 394 lines.

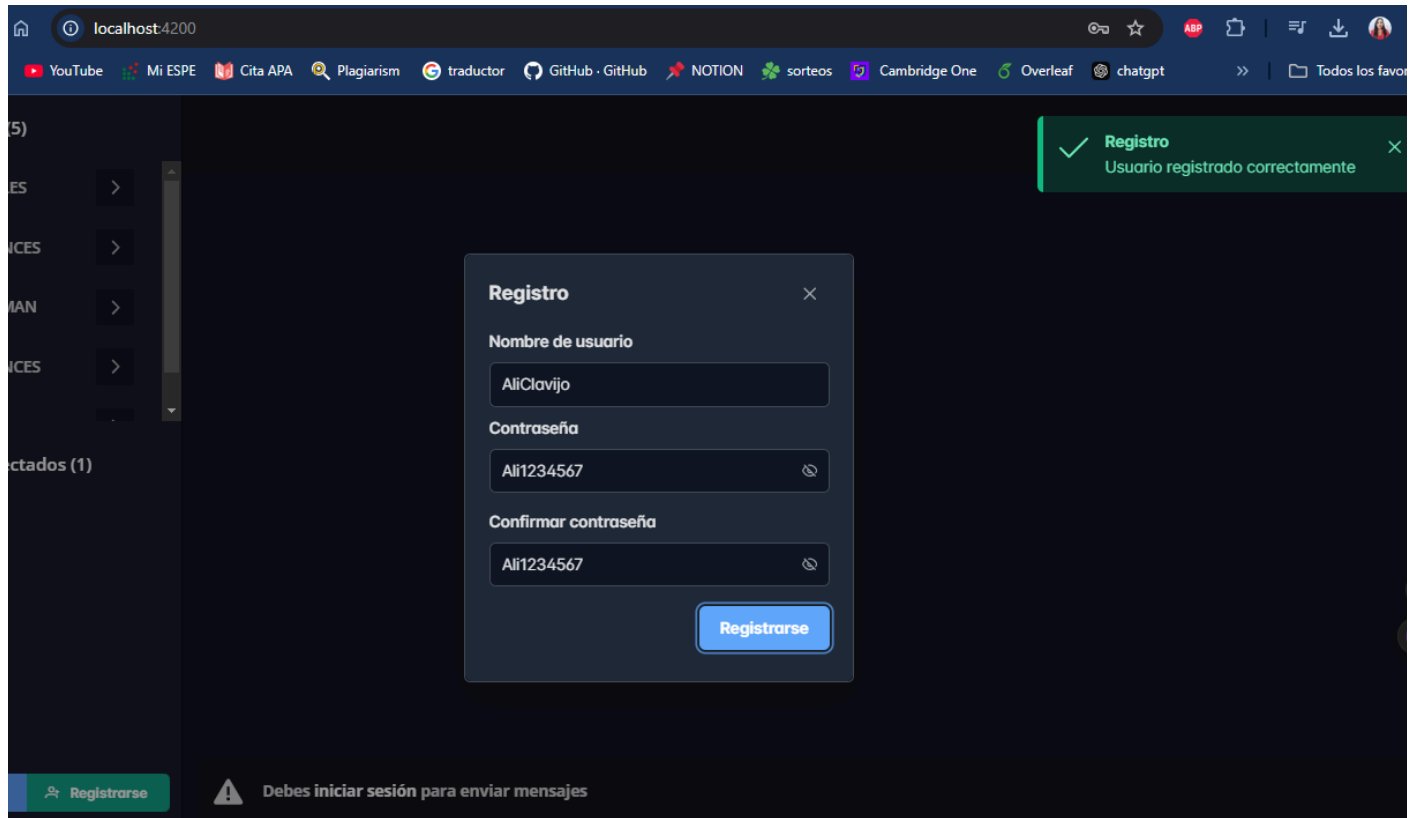




CHATLY

## 12.2 Prueba Dinámica - Selenium IDE

Verificar la funcionalidad del registro de usuario (Usuario y contraseña correcta)





# CHATLY

Selenium IDE - PruebasConstruccion\*

Project: PruebasConstruccion\*

Tests +

Search tests...

Crear Una Sala

✓ Registro\*

▶▶▶▶

http://localhost:4200

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1382x784	
3	✓ click	css= p-button-success > .p-button-label	
4	✓ mouse down	css= .ng-pristine:nth-child(1)	
5	✓ mouse up	css= .text-red-500 > .ng-star-inserter	
6	✓ click	css= .ng-pristine:nth-child(1)	
7	✓ click	id=username	

Command

Target

Value

Description

Log

Reference

7. click on id=username OK 23:39:18

8. type on id=username with value AliClavijo OK 23:39:18

9. click on css=#password .p-inputtext OK 23:39:18

10. Trying to find css=.p-filled:nth-child(1)... OK 23:39:18

11. click on css=#password2 .p-inputtext OK 23:39:35

12. type on css=#password2 .p-inputtext with value Ali1234567 OK 23:39:35

13. click on css=#password .p-icon OK 23:39:36

14. click on css=#password2 .p-icon OK 23:39:37

15. click on css=.flex > .p-element > .p-ripple OK 23:39:38

'Registro' completed successfully 23:39:39



## CHATLY

Verificar la funcionalidad del inicio de sesión

- Llenar el campo de Contraseña, con los datos correctos.

The screenshot shows the Chatly application interface with a login modal open. The modal is titled 'Ingresar' and contains two input fields: 'Nombre de usuario' with the value 'AliClavijo' and 'Contraseña' with the value 'Ali1234567'. A blue 'Ingresar' button is at the bottom right of the modal. In the background, the 'Salas de chat (5)' list includes 'CLASE DE INGLES', 'CLASE DE FRANCES', 'CLASE DE ALEMAN', and 'CLASE DE FRANCES'. The 'Usuarios conectados' section shows a single user 'Admin' with a green status indicator. At the bottom of the screen, there are buttons for 'Ingresar' and 'Registrarse', and a warning message: 'Debes iniciar sesión para enviar mensajes'.

The screenshot shows the Chatly application after a successful login. A green notification banner at the top right reads 'Bienvenido al chat' and 'Bienvenido AliClavijo'. The 'Salas de chat (5)' list on the left remains the same. The 'Usuarios conectados (1)' section now shows the user 'Admin' with a green status indicator. At the bottom, the 'Ingresar' and 'Registrarse' buttons are still present, and the warning message 'Debes iniciar sesión para enviar mensajes' is still displayed.



CHATLY

Selenium IDE - PruebasConstruccion\*

Project: PruebasConstruccion\*

Executing ▾

✓ Inicio de Sesión \*

http://localhost:4200

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1382x784	
3	✓ click	css=.login .p-button-label	
4	✓ type	id=username	AliClavijo
5	✓ click	css=.p-password-input	
6	✓ type	css=.p-password-input	Ali1234567
7	✓ click	css=.p-element:nth-child(2) > .p-icon	

Command:  //

Target:

Value:

Description:

Runs: 1 Failures: 0

Log Reference

Running 'Inicio de Sesión '

1. open on / OK	00:00:46
2. setWindowSize on 1382x784 OK	00:00:47
3. click on css=.login .p-button-label OK	00:00:47
4. type on id=username with value AliClavijo OK	00:00:58
5. click on css=.p-password-input OK	00:00:58
6. type on css=.p-password-input with value Ali1234567 OK	00:00:59
7. click on css=.p-element:nth-child(2) > .p-icon OK	00:00:59
8. click on css=.p-button-label:nth-child(1) OK	00:00:59
'Inicio de Sesión ' completed successfully	00:00:59

- Llenar los campos con datos incorrectos



CHATLY

✕ Error  
Usuario o contraseña incorrectos ✕

Ingresar ✕

Nombre de usuario

Alisson

Contraseña

.....



Ingresar



Debes iniciar sesión para enviar mensajes



CHATLY

Selenium IDE - PruebasConstruccion\*

Project: PruebasConstruccion\*

Tests +

Search tests...

Crear Una Sala

✓ Inicio de Sesión

✓ Inicio de Sesión Incorrecto

http://localhost:4200

	Command	Target	Value
6	✓ type	id=username	Alisson
7	✓ click	css=.p-password-input	
8	✓ type	css=.p-password-input	alisson
9	✓ click	css=.p-button-label:nth-child(1)	
10	✓ mouse over	css=.p-button-label:nth-child(1)	
11	✓ mouse out	css=.p-button-label:nth-child(1)	
12	✓ click	css=.p-toast-icon-close-icon > path	

Command

Target

Value

Description

Log Reference

4. click on id=username OK 00:03:14

5. click on id=username OK 00:03:15

6. type on id=username with value Alisson OK 00:03:16

7. click on css=.p-password-input OK 00:03:17

8. type on css=.p-password-input with value alisson OK 00:03:18

9. click on css=.p-button-label:nth-child(1) OK 00:03:19

10. mouseOver on css=.p-button-label:nth-child(1) OK 00:03:20

11. mouseOut on css=.p-button-label:nth-child(1) OK 00:03:21

12. click on css=.p-toast-icon-close-icon > path OK 00:03:22

'Inicio de Sesión Incorrecto' completed successfully 00:03:23



CHATLY

Crear una Sala Correctamente

localhost:4200

Gmail YouTube Mi ESPE Cita APA Plagiarism traductor GitHub · GitHub NOTION sorteos Cambridge One Overleaf

Admin  
En línea

Salas de chat (3)

CLASE DE INGLES

CLASE DE FRANCES

CLASE DE ALEMAN

Usuarios conectados (1)

Admin

### Crear sala de chat

Nombre:

CLASE DE FRANCES

Descripción:

NUEVA CLASE, GRATIS

Guardar







## 12.3 Prueba de Integración

La prueba contextLoads verifica que el contexto de Spring Boot se cargue correctamente al iniciar la aplicación. Comprueba que el ApplicationContext no sea nulo, lo que garantiza que los componentes y configuraciones necesarias están bien registradas y funcionales. Usa `@SpringBootTest` para inicializar el contexto completo y `AssertJ` para realizar la validación.

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `jwt`, `presenter`, `repository`, `service`, `utils`, `ChatWebSocketServerApplication`, `resources`, `db.changelog`, `application.properties`, `test`, `java`, `ec.edu.espe.chatws.chatwebsocketserver`, `ChatWebSocketServerApplicationTests`, `.gitignore`, and `build.gradle`.
- Editor:** Displays the `ChatWebSocketServerApplicationTests.java` file with the following code:

```
1 package ec.edu.espe.chatws.chatwebsocketserver;
2
3 import ...
4
5 @SpringBootTest
6 class ChatWebSocketServerApplicationTests {
7
8     @Autowired
9     private ApplicationContext applicationContext;
10
11     @Test
12     void contextLoads() {
13         // Verificar que el contexto de Spring se cargue correctamente
14         assertThat(applicationContext).isNotNull();
15     }
16 }
```
- Run Console:** Shows the test results for `ChatWebSocketServerApplicationTests`. The test `contextLoads` passed successfully in 3 seconds and 370 milliseconds. The console output includes:

```
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava
> Task :processTestResources NO-SOURCE
> Task :testClasses
```

The screenshot shows the Run console output for the test `ChatWebSocketServerApplicationTests`. The test `contextLoads` passed successfully in 3 seconds and 370 milliseconds. The console output includes:

```
Run ChatWebSocketServer [:ChatWebSocketServerApplicati... x ChatWebSocketServerApplicationTests
Test Results 3 sec 370 ms Tests passed: 1 of 1 test - 3 sec 370 ms
2024-12-02T23:48:12.340-05:00 INFO 21544 --- [ChatWebSocketSe
2024-12-02T23:48:12.391-05:00 INFO 21544 --- [ChatWebSocketSe
BUILD SUCCESSFUL in 38s
4 actionable tasks: 2 executed, 2 up-to-date
23:48:13: Execution finished ':test --tests "ec.edu.espe.chatw
```