

E1_INF331

admin -> revisar reportes/eliminar contenido.

usuario -> 3ROLES (prestador servicio - cliente-admin), busca servicios...(pueden ser simultáneo).

servicio -> prestar mi servicio de gasfiteria.

CRUD SERVICIO.

- Listar elementos
- Buscar un elemento específico
- Visualizar el detalle de un elemento
- Agregar un nuevo elemento
- Editar un elemento existente
- Eliminar un elemento

TO DO:

-entidad servicios (CRUD)

-terminar usuarios

-Lista de REQ JIRA

-STORYPOINTS JIRA

REQUERIMIENTOS:

FUNCIONALES:

1. El sistema debe permitir que cualquier persona **se registre como usuario** y seleccione ROL

Criterios de aceptación:

1. El sistema debe registrar correctamente la cuenta si todos los datos son válidos.
2. El sistema no permite continuar si falta información obligatoria.
3. El usuario puede seleccionar sólo un rol inicial al finalizar el registro.
4. El sistema debe verificar que la credencial de usuario no esté previamente registrada.

Restricciones y validaciones:

1. El correo debe ser único y válido (formato estándar).
2. El campo nombre debe contener solo letras, diacríticas y espacios.

2. **Creación y publicación de servicios** ofrecidos.

Cada servicio incluirá: 1. Un título 2. Una Categoría 3. Una descripción del servicio 4. Un precio 5. Una imagen (opcional) 6. La zona de atención.

Criterios de aceptación:

1. El sistema permite al prestador crear múltiples servicios desde su panel.
2. El sistema impide publicar un servicio si falta alguno de los campos obligatorios.
3. Al guardar un servicio exitosamente, este queda visible en los resultados de búsqueda.
4. El sistema muestra un mensaje de éxito al publicar correctamente.

Restricciones y validaciones:

1. El precio debe ser un número mayor a 0.
2. La imagen debe estar en formato .jpg o .png y no superar los 5 MB.
3. La zona de atención y la categoría deben seleccionarse desde una lista disponible en el sistema.
4. El título debe tener entre 5 y 100 caracteres.
5. La descripción debe tener entre 10 y 500 caracteres.

3. **Búsqueda de servicios** con filtros.

Los filtros opcionales incluirán: tipo de servicio, categoría, zona de atención, nombre del prestador, rango de precios.

Criterios de aceptación:

1. El sistema muestra resultados que cumplen con todos los filtros aplicados.
2. El usuario puede combinar múltiples filtros en una misma búsqueda.
3. Si no hay coincidencias, se muestra el mensaje: "No se encontraron servicios que coincidan con tu búsqueda".
4. Los resultados se actualizan tras cada búsqueda ejecutada.

Restricciones y validaciones:

1. Los campos de búsqueda no deben quedar vacíos al presionar "Buscar".
2. La zona de atención y la categoría deben seleccionarse desde una lista disponible en el sistema.

4. **Agendamiento de servicios.**

Criterios de aceptación:

1. El usuario puede ver los horarios disponibles y seleccionar uno.
2. El sistema bloquea horarios ya reservados y no disponibles.
3. Tras confirmar la cita, se guarda en el panel del buscador y del prestador.
4. Se envían notificaciones automáticas a ambas partes.

Restricciones y validaciones:

1. No se pueden agendar fechas pasadas.
2. Solo se pueden reservar horarios definidos por el prestador.
3. Un usuario no podrá tener dos citas en la misma franja horaria. (NO TOPES)
4. Solo servicios activos y prestadores habilitados pueden ser agendados.

Resumen del Proyecto: Una descripción concisa que abarque los objetivos, alcance y propósito del proyecto.

Descripción del Trabajo Realizado: Un detalle de las actividades llevadas a cabo, destacando las funcionalidades implementadas y los logros alcanzados.

Tecnologías de aplicación y su relación con las pruebas: Listado de las tecnologías y herramientas utilizadas en el desarrollo, explicando cómo cada una contribuye al proceso de pruebas y aseguramiento de calidad.

Evidencia del Trabajo Realizado: Incluir capturas de pantalla, fragmentos de código, resultados de pruebas y cualquier otro material que demuestre el progreso y los resultados obtenidos.

Estrategia de Pruebas: Descripción de la metodología de pruebas adoptada, tipos de pruebas realizadas (unitarias, de integración, etc.), pruebas de Frontend, Backend, herramientas empleadas y cobertura de pruebas.

Supuestos y dependencias:

Listado de condiciones asumidas durante el Desarrollo: Cualquier suposición hecha que haya influido en las decisiones de diseño o implementación.

Identificación de factores externos que pueden influir en el proyecto:

Dependencias de terceros, limitaciones tecnológicas o cualquier otro factor externo relevante.

Documentación (en Markdown en el mismo repo):

Para asegurar una documentación completa y útil en el repositorio de GitHub, es esencial incluir los siguientes elementos:

Resumen del Proyecto: Una descripción concisa que abarque los objetivos, alcance y propósito del proyecto.

Descripción del Trabajo Realizado: Un detalle de las actividades llevadas a cabo, destacando las funcionalidades implementadas y los logros alcanzados.

Tecnologías de aplicación y su relación con las pruebas: Listado de las tecnologías y herramientas utilizadas en el desarrollo, explicando cómo cada una contribuye al proceso de pruebas y aseguramiento de calidad.

Evidencia del Trabajo Realizado: Incluir capturas de pantalla, fragmentos de código, resultados de pruebas y cualquier otro material que demuestre el progreso y los resultados obtenidos.

Estrategia de Pruebas: Descripción de la metodología de pruebas adoptada, tipos de pruebas realizadas (unitarias, de integración, etc.), pruebas de Frontend, Backend, herramientas empleadas y cobertura de pruebas.

Supuestos y dependencias:

Listado de condiciones asumidas durante el Desarrollo: Cualquier suposición hecha que haya influido en las decisiones de diseño o implementación.

Identificación de factores externos que pueden influir en el proyecto:

Dependencias de terceros, limitaciones tecnológicas o cualquier otro factor externo relevante.

Para estructurar y presentar esta documentación de manera efectiva, se recomienda:

Archivo README.md: Este archivo, ubicado en la raíz del repositorio, debe proporcionar una visión general del proyecto, incluyendo el resumen, instrucciones de instalación, uso básico y enlaces a documentación más detallada.

Wiki del proyecto: Utilizar la wiki de GitHub para documentar en profundidad aspectos específicos del proyecto, como la estrategia de pruebas, supuestos, dependencias y otros detalles técnicos. Las wikis permiten una organización estructurada y son ideales para documentación extensa.

Carpeta de documentación: Crear un directorio específico dentro del repositorio, por ejemplo, docs, para almacenar documentos adicionales, imágenes, diagramas y cualquier otro material relevante.

TEST:

Unitarios; por clases de backend.

CitaServicesImplTest.java

Nombre del test	Qué valida:
crearCita_atributosOK_deberiaCrearCita()	Se crea la cita con idPrestador y idBuscador y fecha válidas.
crearCita_atributosConflicto_deberiaLanzarExcepcion()	Crear una cita con categoría errónea debiese pasar a excepción y mensaje.
obtenerCitaPorId_llamado_deberiaRetornarCitaExistente()	Buscar una cita por id exitosa.
obtenerCitaPorId_noExiste_deberiaRetornarNull()	Buscar una cita por id inexistente.
actualizarCita_atributosValidos_deberiaActualizarCampos()	Actualizar una cita con id existente la actualiza correctamente.
obtenerCitas_porBuscador_deberiaListarCitas()	Buscar las citas por id de usuario buscador.
obtenerCitas_porPrestador_deberiaListarCitas()	Buscar las citas por id de usuario prestador.
actualizarEstadosDeCitas_pasoFecha_deberiaActualizarPendientesACompletadas()	Citas pasadas deben ser marcadas como completadas.

Pruebas Enpoints:

Crear usuario BUSCADOR:

Conectados / Crear Usuario Buscador

POST

http://localhost:8080/auth/register

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautiful

1

2

3

4

5

6

{

"nombre": "Maria Mella",

"correo": "maria.buscador@gmail.com",

"contrasena": "1234",

"rol": "BUSCADOR"

}

Body

Cookies

Headers (8)

Test Results

409 Conflict

179 ms

309 B

Save Response

JSON

Preview

Visualize

1

2

3

{

"message": "Ya existe una cuenta con este correo"

}

Conectados / Crear Usuario Buscador

SaveShare

POSThttp://localhost:8080/auth/registerSend

ParamsAuthorizationHeaders (8)BodyScriptsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1 {
2   "nombre": "Maria Mella",
3   "correo": "maria.buscador2@gmail.com",
4   "contrasena": "1234",
5   "rol": "BUSCADOR"
6 }
```

BodyCookiesHeaders (8)Test Results200 OK238 ms468 BSave Response

JSONPreviewVisualize

```
1 {
2   "id": 11,
3   "nombre": "Maria Mella",
4   "correo": "maria.buscador2@gmail.com",
5   "contrasena": "1234",
6   "imagen": null,
7   "rol": "BUSCADOR",
8   "zonaAtencion": null,
9   "categoria": null,
10  "descripcion": null,
11  "disponibilidad": null,
12  "permisos": null
13 }
```