

# **APLICATIVO DE HERRAMIENTA DIGITAL PARA DISEÑO DE MOLDES DE INYECCIÓN DE CAUCHO**

**Fecha:** 12/ 07/ 2024

**Autor:** Esteban Garcia Zuluaga

**Propietario:**

**INDUSTRIAS CADI S.A.S**

# CONTENIDO

APLICATIVO DE HERRAMIENTA DIGITAL PARA DISEÑO DE MOLDES DE INYECCIÓN DE CAUCHO .....	1
INTRODUCCIÓN .....	3
<b>OBJETIVO DEL DOCUMENTO</b> .....	3
<b>ÁMBITO DEL PROYECTO</b> .....	3
<b>PUBLICO OBJETIVO:</b> .....	3
DESCRIPCIÓN DEL PROYECTO .....	3
<b>OBJETIVOS DEL PROYECTO:</b> .....	3
<b>FUNCIONALIDADES Y USABILIDAD:</b> .....	3
MANUAL DE USUARIO: .....	3
<b>REQUISITOS DEL PROYECTO:</b> .....	4
ARQUITECTURA DEL SISTEMA .....	4
<b>DESCRIPCIÓN GENERAL DE LA ARQUITECTURA:</b> .....	4
<b>COMPONENTES DEL SISTEMA:</b> .....	5
DISEÑO DEL SISTEMA.....	5
<b>DIAGRAMA UML:</b> .....	5
<b>DISEÑO DE LA BASE DE DATOS:</b> .....	5
<b>INTERFAZ DE USUARIO (UI/UX)</b> .....	6
<b>DISEÑO DETALLADO DE COMPONENTES:</b> .....	12
GUÍA DEL DESARROLLADOR: .....	12
PRUEBAS Y VALIDACIÓN .....	17
<b>PLAN DE PRUEBAS:</b> .....	17
<b>CASOS DE PRUEBA:</b> .....	18
MANTENIMIENTO .....	23
<b>PROCEDIMIENTOS DE MANTENIMIENTO:</b> .....	23
CONCLUSIONES .....	25
<b>RESUMEN DEL PROYECTO:</b> .....	25
<b>RECOMENDACIONES FUTURAS:</b> .....	26
ANEXOS .....	27
<b>GLOSARIO DE TÉRMINOS</b> .....	27

# **INTRODUCCIÓN**

## **OBJETIVO DEL DOCUMENTO**

Este documento se relaciona toda la información detallada del proyecto; Podrás encontrar el alcance, descripción, arquitectura del sistema, plan de mantenimiento, pruebas realizadas al aplicativo, documentación relacionada al código y un manual de usuario y una guía de desarrollador.

## **ÁMBITO DEL PROYECTO**

En relación a la necesidad de tener una transformación tecnológica en la compañía se busca implementar nuevas herramientas que lo permitan; Como un aplicativo para el diseño de moldes de inyección de caucho, que a su vez, centraliza datos, gestiona información y genera funcionalidades en el aplicativo en base a los datos.

## **PÚBLICO OBJETIVO:**

Desarrolladores, clientes, usuarios finales.

## **DESCRIPCIÓN DEL PROYECTO**

### **OBJETIVOS DEL PROYECTO:**

Conforme con algunas necesidades de la compañía se desarrolla una solución que suple estas, en relación a los requerimientos de los usuarios, tales como la digitalización de la información, gestión de los datos, transformación de la información, interacción con el diseñador para acelerar el diseño de moldes e interacción tecnológica con los clientes y usuarios.

### **FUNCIONALIDADES Y USABILIDAD:**

### **MANUAL DE USUARIO:**

Se crean varios tipos de funcionalidades tales como, buscar productos por dimensiones y crear, gestionar archivos, y registro y login de usuarios.

Se gestiona la información y se hace una transformación de los datos agrupándolos en tablas con cada uno de sus atributos tales como: Descripciones, dimensiones, identificadores, compuesto, tamaño de moldes, tipo de distribución, peso, y documentos relacionados a sus códigos, entre otras características.

La información que este aplicativo brinda al usuario se es gracias a su capacidad de gestionar y manipular los datos.

Para lograr acceder a esta, solo hace falta que el usuario ingrese a “**Buscar O-rings/Arandelas**” digite las dimensiones del producto que desea buscar, donde también tiene una calculadora que le ayuda a buscar un aproximado si tiene alguna dimensión faltante.

Para el registro de la información se tiene un formulario llamado “**crear O-ring/Arandela**” En este se precisa de la información requerida mínima para crear un nuevo producto que se almacenará en la DB, donde tiene un modelo de creación de productos y almacenamiento de los mismos, desde este formulario se podrá actualizar la información de productos ya existentes agregándoles una nota en la descripción si así lo desea.

Esto solo funcionará con aquellos usuarios que hallan sido creados con el atributo de “**Administrador**” por un “**Administrador**” esto para evitar el registro/actualización indevido de la información almacenada.

Para crear usuarios solo se le permite crear por aquellos usuarios que hallan sido creados con el atributo de “**Administrador**” estos podrán proporcionar/inhibir permisos a los roles de “**Usuario**”. Donde al ser usuarios registrados tienen información más detallada de los productos que consulten.

Con usuarios “**No registrados**” pueden acceder al apartado de “**/customersHome**” y “**Buscar O-rings/Arandelas**” y acceder a información básica de los productos.

## REQUISITOS DEL PROYECTO:

**Funcionales:** Integración de datos, generación de consultas, búsqueda de productos y centralización de documentación.

**No funcionales:** Rendimiento, escalabilidad, seguridad, integración con API's.

## ARQUITECTURA DEL SISTEMA

### DESCRIPCIÓN GENERAL DE LA ARQUITECTURA:

Se desarrolla una arquitectura cliente-servidor se basa en el concepto de la existencia de un servidor (proveedor del servicio) con una interfaz de usuario web y un backend que maneja la lógica de negocio y las consultas sean “request” o “query”.

Esta arquitectura nos ayuda a centralizar todos los recursos dispobles que hay en un unico punto lo que hace una administración más sencilla y mas robusto para el uso de acciones dañinas, a su vez es escalable y funciona independiente cada servicio, lo que lo hace más sencillo para mejorar y dar mantenimiento a cada pieza de la arquitectura

COMPONENTES DEL SISTEMA:

Los componentes centrales del sistema son:

**Base de datos:** Desarrollada en MongoDB centralizada para almacenar los datos de cada producto, usuario y documentos.

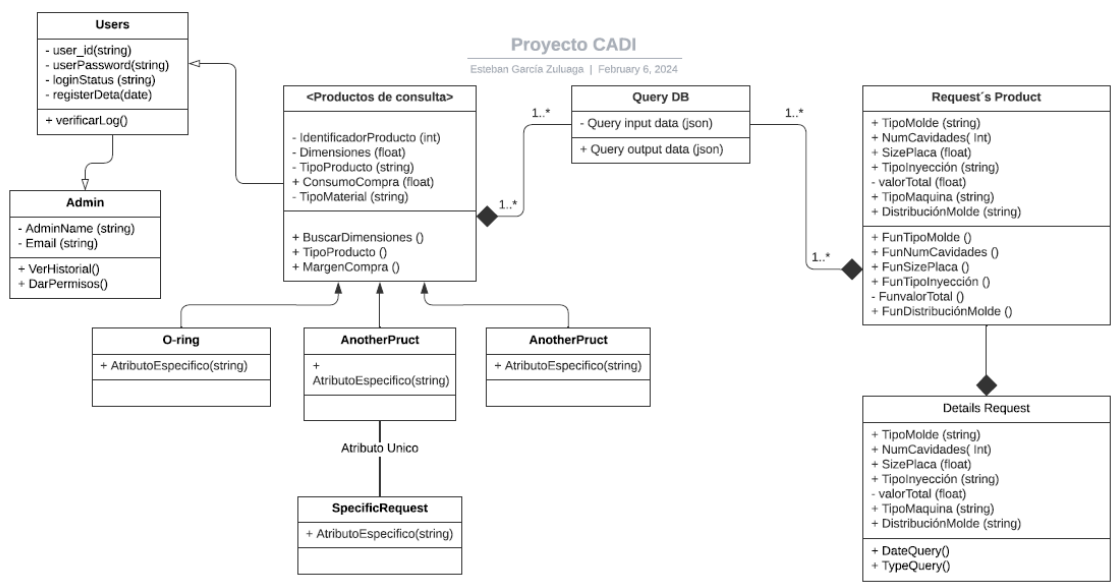
**Back-end:** Conformada por numerosas Apis y frameworks ayudan al desarrollo de la lógica de negocio, tales como una API RESTful con **axios** encargada de generar endpoints entre el front-end y el back-end para la interacción con los servicios web.

Integración con **bcryptjs** framework encargado de generar una encriptación de la información de los usuarios, entre otras librerías como **cors**, **express**, **mongoose**, **cookie-parser**, **dotenv**, **helmet**, **jsonwebtoken**, **mongodb**, **morgan**, **multer** y **zod** que conforman los componentes del sistema.

**Front-end:** Conformar la interacion visual con el cliente con frameworks como **react**, **react dom**, **react-hook-from** y **react-router-dom**.

DISEÑO DEL SISTEMA

DIAGRAMA UML:



DISEÑO DE LA BASE DE DATOS:

El diseño de la base de datos MongoDB almacena documentos, se identifican las colecciones necesarias y los índices que se puedan requerir para optimizar las consultas

Campo	Tipo de Datos	Descripción
_id	ObjectId	Identificador único del documento
ID	String	Identificador del producto
DESCRIPCIÓN	String	Descripción del producto
CodigoCompu	String	Código de computadora
Medidas	String	Medidas del producto
Espesor	Number	Espesor del producto
Dexterno	Number	Diámetro externo
Dinterno	Number	Diámetro interno
Peso_gr	Number	Peso en gramos
PrecioProducir	Number	Precio de producción
PrecioV	Decimal128	Precio de venta
IDplano	String	Identificador del plano
Idmolde	String	Identificador del molde
Mtamaño	String	Tamaño del molde
Ncavidades	String	Número de cavidades
Nplacas	String	Número de placas
Tdistribución	String	Tipo de distribución
Patin	String	Indicador de patín (Sí/No)
T_Maquina	String	Tipo de máquina
TMolde	String	Tipo de molde
TProceso	String	Tipo de proceso

## INTERFAZ DE USUARIO (UI/UX)

Las interfases desarrolladas se enfocan en una UX y UI agradable y que mejora la experiencia y las sensaciones del usuario cuando utiliza el producto.

**Interfaz del home page:** En esta página se puede navegar hacia páginas de log-in, customers Home donde el usuario no registrado puede hacer búsqueda de productos, y el usuario registrado crear productos y gestión de permisos.



**Customers Home:** Esta pagina permite a los usuarios no registrados hacer consultas de productos tales como arandelas y o-rings,



**Customer QueryOring:** Esta pagina al igual que **Customer QueryArandela** permite al usuario hacer calculo de las medidas faltantes, las consultas y ver la respuesta de esta.

INDUSTRIAS

CADI S.A.S

PRODUCTOS PARA EL DISEÑO DE MOLDES DE INYECCIÓN DE CAUCHO

Query Products Manager

Recargar Página

Login

¡Consulta realizada con éxito!

CONSULTAR O-RINGS

2.3

8

11

GENERAR CONSULTA

Calcular valor de O-ring

I/D

O/D

W

Calcular

Limpiar

Resultado de la consulta

Descripción	Código	W m.m	I/D m.m	O/D m.m	Peso gr
ORING MILIMETRICO 2X7,5X11,5 EPDM	26011	2	7.5	11.5	
ORING MILIMETRICO 2X8X12 EPDM	26012	2	8	12	
ORING 011	00011	1.78	7.65	11.21	
O'RING 011 39090 VITON	10611	1.78	7.65	11.21	
ORING MILIMETRICO 2,00 - 7,50 - 11,50 VITON	22012	2	7.5	11.5	

Paginas

1 2 3 4

**Login page:** Permite ingresar solo usuarios previamente registrados.

INDUSTRIAS

CADI S.A.S

PRODUCTOS PARA EL DISEÑO DE MOLDES DE INYECCIÓN DE CAUCHO

Query Products Manager

Recargar Página

Login

Accede

Iniciar sesión para continuar

Email

Password

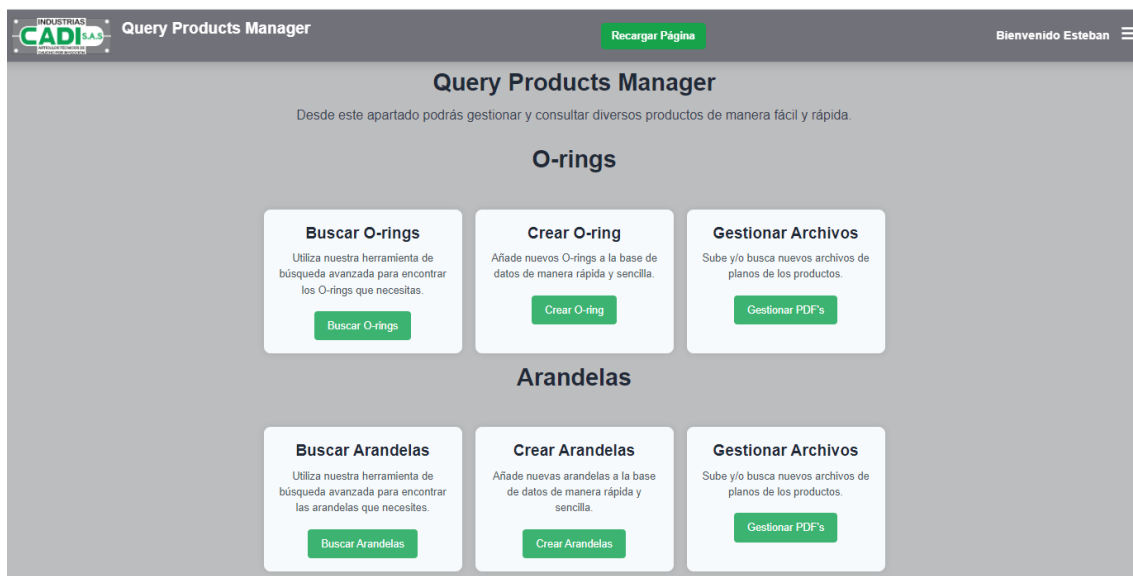
Login

Si no estas registrado solicita un usuario

**Products Home:** Permite al usuario navegar por las diferentes paginas de los productos, sea o-ring, arandelas.

También se puede navegar entre las diferentes páginas mediante el **LOGO** de CADI que te redirecciona a la página principal y **Query Products Manager** te redirecciona a la página principal de los productos.





**QueryO-ring/query:** Esta pagina nos permite hacer consultas donde podemos calcular las dimensiones de los productos. Al igual que **queryArandela/query** y también podemos ver la respuesta de la consulta con más información en comparación con los resultados de las consultas para los clientes.

The screenshot shows the 'CONSULTAR O-RINGS' form within the 'Query Products Manager' interface. At the top, there's a green notification bar that says '¡Consulta realizada con éxito!'. Below this, the form has three input fields for dimensions: '2.3', '8', and '11'. A green 'GENERAR CONSULTA' button is positioned below these fields. Underneath, there's a section titled 'Calcular valor de O-ring' with three input fields labeled 'I/D', 'O/D', and 'W'. At the bottom of this section are two buttons: 'Calcular' (grey) and 'Limpiar' (red). The interface also includes a 'Recargar Página' button and a 'Login' link in the top right corner.

**Resultado de la consulta:** Esta tabla nos da información relevante del producto como su descripción e identificador, compuesto, medidas, información relevante del molde como sus medidas, número de placas y un identificador para el plano y/o documento.

INDUSTRIAS

CADI

S.A.S

soluciones industriales

Query Products Manager

Recargar Página

Bienvenido Esteban

Resultado de la consulta

Descripción	Codigo	Compuesto	W m.m	I/D m.m	O/D m.m	ID mold	T Mold cm	# Cavidades	# Placas	Patin	Peso gr	Precio P	Tipo maquinas
ORING MILIMETRICO 2X7,5X11,5 EPDM	26011	No-Registrado	2	7.5	11.5	M26011	30X30	180	2	0		0	
ORING MILIMETRICO 2X8X12 EPDM	26012	No-Registrado	2	8	12	M26012	0	110	0	0		0	
ORING 011	00011	3-1-1-69	1.78	7.65	11.21	M11	35X35	228	2	NO		12.32	
O'RING 011 39090 VITON	10611	No-Registrado	1.78	7.65	11.21	M10611	30X30	228	2	0		0	
ORING MILIMETRICO 2,00 - 7,50 - 11,50 VITON	22012	No-Registrado	2	7.5	11.5	M22012	30X30	180	2	0		0	

1

2

3

4

**Crear productos:** Este formulario nos permite crear productos para O-rings, y en esta dirección [/queryArandela](#) para la creación de Arandelas.

El requerimiento de datos para la creación de productos es acorde al modelo desarrollado para almacenar productos en la DB estos solo los podrán crear aquellos usuarios que tengan un rol de “**Administrador**” dándole al botón guardar enviará el producto creado a la DB.

INDUSTRIAS CADI S.A.S.		Query Products Manager										Recargar Página													
												Bienvenido Esteban													
SUBIR DATOS PARA O-RINGS																									
Codigo CADI																									
Descripción																									
Codigo del Compuesto																									
Espesor																									
Diametro Externo																									
Diametro Interno																									
Peso en gramos																									
Precio Producir																									
Tipo de molde																									
C																									
Tipo de Proceso																									
Inyección																									
Guardar																									

**Gestionar Archivos:** Esta página /pdfPage permite a los usuarios administrar los archivos que esten relacionados con los productos. Para subir un archivo se nombra en base al producto que tiene relación con el ID de CADI,y seleccionar el documento desde la carpeta desplegable.

Para buscar solo debes usar el identificador con el que se subió o con el identificador que te arroja en la tabla de consultas y te abrirá los documentos en una pestaña nueva.

The screenshot shows the 'Query Products Manager' interface. At the top, there is a header with the CADI S.A.S. logo, the title 'Query Products Manager', a 'Recargar Página' button, and a user greeting 'Bienvenido Esteban' with a menu icon. The main content area features two forms. The first form, titled 'Subir archivo PDF', includes an input field for 'ID del plano', a dropdown menu labeled 'Seleccionar archivo' with the text 'Sin archivos seleccionados', and a green 'SUBIR ARCHIVO' button. The second form, titled 'Buscar archivo PDF', includes an input field for 'ID del plano' and a green 'BUSCAR ARCHIVO' button.

**Menú desplegable:** El menú desplegable permite navegar de una forma más rápida por las diferentes páginas y cerrar la sesión.

The screenshot shows the 'Query Products Manager' interface with the 'O-rings' section. The header is the same as the previous screenshot. Below the header, there is a subtitle 'Desde este apartado podrás gestionar y consultar diversos productos de manera fácil y rápida.' The main content area is titled 'O-rings' and contains three cards. The first card, 'Buscar O-rings', describes an advanced search tool and has a green 'Buscar O-rings' button. The second card, 'Crear O-ring', describes adding new O-rings to the database and has a green 'Crear O-ring' button. The third card, 'Gestionar Archivos', describes uploading or searching for new product plans and has a green 'Gestionar PDF's' button. On the right side, a dropdown menu is open, showing options: 'Buscar Productos', 'Crear Productos', 'Buscar/Subir PDF', 'Registrar usuarios', and 'Cerrar sesión'.

**Crear usuarios:** La creación de usuarios solo tienen acceso aquellos usuarios que tengan como rol “Administrador”. En la ruta /register permite crear usuarios donde se debe darle un correo, username, contraseña mínimo 6 caracteres y seleccionar un **ROL** en base a los permisos que le quieras dar al usuario.

INDUSTRIAS CADI S.A.S. Query Products Manager Recargar Página Bienvenido Esteban

Crear una cuenta nueva

Username

Email

Password

Selecciona un rol:

Usuario

Registrar Usuario nuevo

## DISEÑO DETALLADO DE COMPONENTES:

### GUÍA DEL DESARROLLADOR:

La implementación de cada uno de estos componentes conforma el script de la aplicación desarrollada, cada componente se refiere a una funcionalidad o microservicio que se conectan entre sí, para permitir una sincrónica funcionalidad del APLICATIVO DE HERRAMIENTA DIGITAL PARA DISEÑO DE MOLDES DE INYECCIÓN DE CAUCHO. Esto se conforma de dos componentes principales los cuales son el back-end y front-end.

#### Back-end:

**Src:** Contiene todos los componentes lógicos que conforman el back-end, la configuración con la DB, y un archivo index.js que es el iniciador del aplicativo.

**Controllers:** Este componente se encarga de la creación de las funciones que se van a ejecutar cuando el usuario visite una URL.

Este componente tiene los controladores de las Arandelas y O-rings el cual contiene funciones CRUD que permiten consumir y manipular la información que solicite el usuario, llamando y almacenando información en la DB.

El componente **auth.controller** esta encargado del registro y manipulación del login y logout de los usuarios y encargado de encriptar, las contraseñas y proteger rutas por medio del middleware.

**Files:** Este componente guarda todos los documentos (planos) de los productos

**Libs:** Este componente maneja una función que genera tokens para el login, registra/sing up.

**Middlewares:** Este componente se encarga de proteger rutas, validar schemas para la creación de usuarios, protección de rutas en base a un rol, autenticación de usuarios para filtrar quien puede hacer solicitudes al back-end.

El archivo **multer** es un framework que permite el manejo de documentos en la DB y en la UI.

**Models:** Este componente se encarga de los archivos que contienen un schema o modelos de la creación de productos, registro de usuarios, y para la creación de roles. Este modelo se valida con los archivos del componente middlewares

**Routes:** Este componente crea las URLs del back-end, define endpoints, para las solicitudes del front.

Tanto las arandelas, O-rings y usuarios tienen definidas las rutas en este componente donde se integran con los middlewares para la validación de los schemas, autenticación requerida, rol y llama las funciones del componente **controllers** en cada ruta para hacer las solicitudes.

**Schemas:** Este componente se encarga de analizar y validar los datos que llegan al Back-end.

Tanto las Arandelas, o-rings y usuarios tienen un schema de validador de datos recibidos del Front-end. Estos schemas tienen un mínimo de requerimientos para permitir interacción de la información.

**App.js:** Este archivo se encarga de definir las rutas y la lógica de la aplicación en conjunto con el archivo **Index.js**.

**Package.json:** Están las dependencias y el comando para iniciar el servidor “npm run dev”.

## Front-end:

**Src:** Contiene todos los componentes lógicos y visuales que conforman el front-end, la configuración con la DB, las páginas de navegación y un archivo App.jsx que es el iniciador del aplicativo.

**Api:** Este componente contiene los endpoints generados con el framework **axios**, los cuales se dividen los archivos `auth.js` para el registro, login y verificación del usuario, este genera la comunicación cliente servidor.

Los archivos `queryOrings.js` y `queryArandelas.js`: Estos contiene la rutas y endopints para el registro, manipulación, actualización y solicitud de productos.

**Assest:** Este componente almacena las imágenes que consume la aplicación.

**Components:** Este componente contiene los estilos de la barra de navegación y la lógica de esta, la cual se encarga agilizar la navegación en el aplicativo y contiene un “`useState`” que en relación a la autenticación de un usuario tiene diferentes funcionalidades.

**Context:** Este componente contiene el contexto tanto para los productos como para el registro y login de usuarios. Esta es una manera de pasar datos a través del árbol de componentes sin tener que pasar propiedades manualmente en cada nivel.

El contexto se encarga de los datos que deben ser accesibles por muchos componentes a diferentes niveles de la jerarquía, como datos de autenticación, tema, o configuración global.

El código se utiliza para manejar la lógica de negocio relacionada con las arandelas, O-rings y usuarios, incluyendo la creación y consulta de productso y usuarios, para compartir esta lógica y datos entre los componentes de la aplicación sin necesidad de pasar propiedades a través de múltiples niveles de componentes.

Este enfoque permite que el estado y la lógica relacionados con las distintas funcionalidades del aplicativo se compartan entre múltiples componentes sin necesidad de pasar props manualmente.

**Pages:** Este componente almacena todos los archivos relacionados con las diferentes paginas que conforman el aplicativo, desde los home page, las paginas para de cada funcionalidad, como registrar productos, obtener, login, registrar usuarios, página de documentos.

Estos archivos se encargan de la interacción con el cliente y como se hace la solicitud al servidor.

Este componente React proporciona una interfaz de usuario para buscar en una base de datos los productos según sus dimensiones y realizar cálculos básicos para encontrar valores dimensionales faltantes.

Los archivos **ArandelasPage** y **QueryOringPage** se encargan de la obtención inicial de datos y manejo de errores:

El componente obtiene datos al montarse usando el hook `useEffect` y la función `getQueryArandela/ getQueryOring` de un contexto externo (componente anterior)

Establece un estado de carga mientras se obtienen los datos y maneja posibles errores durante la obtención.

La entrada del usuario y validación del formulario:

El componente usa `React Hook Form (useForm)` para administrar la entrada del usuario para el ancho (W), diámetro interno (Dinterno) y diámetro externo (Dexterno) de la arandela. Este valida la entrada del usuario para asegurarse de que todos los campos obligatorios estén llenos y que los valores sean números válidos.

El componente para calcular los productos proporciona una sección para calcular la dimensión faltante de los productos (DI, DE o CS) en base a los otros dos valores proporcionados.

El usuario ingresa los valores conocidos y el componente calcula el faltante usando una función `handleCalculate`.

La función `getQueryArandela/getQueryOring` consulta una base de datos externa según las dimensiones proporcionadas por el usuario. Los datos obtenidos (información de los productos) se almacenan en la variable de estado `medidasArandela/ medidasOring`.

El componente muestra un mensaje de error si no se encuentran productos para los criterios dados. También implementa la paginación para mostrar los resultados de búsqueda en partes.

El componente renderiza un formulario para que los usuarios ingresen las dimensiones de la arandela y un botón de envío para iniciar la búsqueda. Muestra los resultados de la búsqueda en una tabla y proporciona funcionalidades para calcular las dimensiones faltantes y navegar a través de los resultados paginados.

Para la creación de los productos los archivos **CreateArandelasPage** y **CreateOringPage**

Utilizan un componente `React` llamado `CreateArandelasPage / CreateOringPage` (Página de Crear Arandelas/Orings) que se utiliza para crear nuevas entradas para (los productos) en una base de datos.

Formulario para la entrada del usuario:

El componente renderiza un formulario con varios campos de entrada para que los usuarios ingresen detalles sobre la arandela, incluyendo:

ID (obligatorio)  
Descripción (obligatorio)  
Material (obligatorio)  
Espesor (W) (obligatorio)  
Diametro interno (Dinterno) (obligatorio)  
Diametro externo (Dexterno) (obligatorio)  
Peso (Peso)  
Tamaño del molde (MoldTamaño)  
Número de cavidades (Ncavidades)  
Número de placas (Nplacas)  
Tipo de distribución (Distribucion)  
¿Lleva patín? (Patin)  
Línea del producto (Linea)  
Tipo de máquina (Tmaquina)  
Tipo de proceso (TProceso)  
Nombre del archivo PDF (PDF) (obligatorio)

Se hace una validación del formulario:

El componente usa React Hook Form (useForm) para administrar las entradas del usuario y manejar la validación. Se encarga de que todos los campos obligatorios estén llenos y que los campos numéricos (W, Dinterno, Dexterno y rango de PDF) contengan números válidos.

El envío del formulario lo maneja la función onSubmit usando handleSubmit de React Hook Form. Se verifica si alguno de los campos obligatorios está vacío o si los valores numéricos no son válidos. De ser así, arroja un error. Si los datos son válidos, llama a la función createQueryArandela / createQueryOring (para un contexto externo) para crear una nueva entrada de arandela/O-ring en la base de datos.

Mientras se crea la nueva entrada, el componente muestra un indicador de carga. En caso de errores durante la creación de datos (createQueryArandelaErrors/ createQueryOringErrors), muestra los mensajes de error recibidos. Al crear con éxito, muestra un mensaje de éxito.

El componente muestra un encabezado "subir datos de las arandelas/o-rings " (Cargar datos de las arandelas/Orings).

Renderiza un formulario con campos de entrada para varios detalles los productos.



El formulario usa etiquetas claras y concisas para identificar cada campo de entrada.

Los mensajes de error se muestran debajo de los campos de entrada correspondientes en texto rojo.

El botón verde "Guardar" (Guardar) está disponible para enviar el formulario y crear una nueva entrada de arandela/Oring.

Los mensajes de éxito y error se muestran después del envío del formulario.

Este componente React proporciona una interfaz de usuario para crear nuevas entradas de productos en una base de datos. Incluye funcionalidades para la entrada del usuario, validación de formulario, envío de datos y manejo de errores.

El componente **LoginPage** es una página de inicio de sesión en React que utiliza las librerías react-hook-form para manejar formularios y react-router-dom para la navegación.

Este componente permite a los usuarios iniciar sesión y maneja tanto la validación de entradas como la autenticación, incluyendo la redirección en caso de éxito.

El componente **RegisterPage** permite a los usuarios crear una nueva cuenta proporcionando un nombre de usuario, correo electrónico, contraseña y rol. Se utiliza react-hook-form para la gestión y validación de los campos del formulario.

Muestra mensajes de error si los campos obligatorios no se completan o si hay errores de registro. Llama a la función signup del contexto de autenticación con los datos del formulario.

La página de inicio presenta tres opciones principales: buscar productos, crear productos y gestionar permisos de usuario. Utiliza Link de react-router-dom para proporcionar navegación interna sin recargar la página.

Aplica estilos personalizados importados desde Home.css para mejorar la apariencia visual del componente.

## PRUEBAS Y VALIDACIÓN

### PLAN DE PRUEBAS:

Versión: 1.0

## Introducción

El propósito de este plan de pruebas es delinear la estrategia y el enfoque para probar la aplicación **HERRAMIENTA DIGITAL PARA DISEÑO DE MOLDES DE INYECCIÓN DE CAUCHO**. Este plan define el alcance de las pruebas, los objetivos y los recursos.

**Pruebas funcionales:** Verifica que la aplicación cumpla con todos los requisitos funcionales especificados en el documento de requisitos.

**Pruebas no funcionales:** Evalúa el rendimiento, la usabilidad, la seguridad y otros requisitos no funcionales de la aplicación.

**Pruebas de integración:** Asegurar que la aplicación se integre perfectamente con otros sistemas y componentes.

**Pruebas de sistema:** Validar que todo el sistema cumpla con todos los requisitos especificados.

## Objetivos

- Identificar y prevenir defectos del aplicativo.
- Asegurar que el aplicativo cumpla con todos los requisitos.
- Entregar un producto de alta calidad que cumpla con las expectativas del usuario.

## Definiciones

**Error:** Un defecto de software que hace que la aplicación funcione mal o se comporte de forma inesperada.

**Defecto:** Un término general para cualquier problema con el software.

**Caso de prueba:** Un conjunto de pasos diseñados para probar una característica o funcionalidad específica del aplicativo.

**Conjunto de pruebas:** Una colección de casos de prueba relacionados.

**Entorno de pruebas:** La configuración de hardware y software utilizada para las pruebas.

## CASOS DE PRUEBA:

Las pruebas son esenciales para garantizar que los componentes de React funcionen correctamente, sean confiables y brinden una buena experiencia de usuario en general.

### **Pruebas unitarias:**

Componentes de registro, obtención de productos, login, registro de usuarios y home page:

Se verifica que los componentes manejen correctamente las credenciales de inicio de sesión válidas e inválidas el envío y solicitud de datos

Se asegura que los mensajes de error se muestren apropiadamente para entradas incorrectas.

Se prueba que el componente redirija a la página esperada tras un inicio de sesión exitoso.

Se verifica que register sea una función que se pueda usar para registrar campos de formulario.

Se verifica que handleSubmit sea una función que se pueda usar para manejar el envío del formulario.

Se verifica que formState sea un objeto que contenga información sobre el estado del formulario, incluido un objeto errors que contenga errores de validación para cada campo registrado.

Se verifica que **getQueryArandela** sea una función que se pueda usar para obtener datos de arandelas.

Se verifica que getQueryOrings sea una función que se pueda usar para obtener datos de los O-rings.

Se verifica que medidasArandela sea un array que contenga los datos de las arandelas recuperadas.

Se verifica que errors sea un array que contenga cualquier error que haya ocurrido durante la recuperación de datos.

Se simula la recuperación exitosa de datos llamando a getQueryArandela y configurando medidasArandela con datos de muestra.

Se simula un error de recuperación de datos configurando getQueryArandelasErrors y getQueyOringErros con un mensaje de error simulado.

Se verifica que setLoading se llame para indicar el estado de carga.

Se verifica que setIsDataFetched se llame para indicar si se han recuperado los datos.

Se simula el envío del formulario con valores vacíos para los campos requeridos. Verifique que se devuelvan los errores de validación apropiados en el objeto errors de formState.

Se simula el envío del formulario con valores no numéricos para los campos W, Dinterno y Dexterno.

Se verifica que se devuelva un error que indique que las entradas deben ser números válidos.

Se simula el envío del formulario con datos válidos.

Se verifica que se llame a setLoading para indicar el estado de carga.

Se verifica que se llame a getQueryArandela/getQueryOring con los datos del formulario.

Se verifica que se llame a setIsDataFetched para indicar que se han recuperado los datos.

Se verifica que se manejen los errores lanzados durante la recuperación de datos.

Se verifica que DE se calcule correctamente y se almacene en calculatedValues.

Se verifica que CS se calcule correctamente y se almacene en calculatedValues.

Se simula el ingreso de un valor para DE o CS y deje el otro vacío.

Se verifica que el valor ingresado se almacene en calculatedValues y que el otro valor permanezca sin cambios.

Se simula el cambio de un campo de entrada utilizado para el cálculo.

Se verifica que el valor correspondiente en calculatedValues se actualice con el nuevo valor ingresado.

Se simula el ingreso de valores para los campos de cálculo.

Se verifica que los valores calculados se almacenen en calculatedValues.

Se verifica que todos los valores en calculatedValues se borren.

Se simula tener una gran cantidad de elementos en medidasArandela.

Se verifica que currentItems contenga los elementos de la página actual según el currentPage y itemsPerPage.

Se simula hacer clic en un número de página diferente.

Se verifica que currentPage se actualice y currentItems muestre los elementos de la nueva página.

Se verifica que todos los campos obligatorios (ID, descripción, compuesto, W, Dexterno, Dinterno, PDF) estén marcados con la regla de validación required.

Se envía el formulario con campos obligatorios vacíos y compruebe si se muestran los mensajes de error correspondientes.

Se envía el formulario con valores no numéricos para los campos que deberían ser números (W, Dexterno, Dinterno).

Se verifica que se muestre un mensaje de error indicando que los valores deben ser números válidos.

Se simula la función createQueryArandela para que devuelva una respuesta exitosa.

Se envía el formulario con datos válidos y compruebe si se muestra el mensaje de éxito, el estado de carga se establece en falso (false) y se borran los mensajes de error anteriores.

Se verifica que los datos del formulario se pasen correctamente a la función createQueryArandela.

Se simula la función createQueryArandela para que genere un error.

Se envía el formulario con datos válidos y compruebe si se muestra el mensaje de error, el estado de carga se establece en falso (false) y no se muestra el mensaje de éxito.

Se verifica que el mensaje de error esté relacionado con el error generado por la función createQueryArandela.

Se envía el formulario con campos obligatorios vacíos y compruebe si los mensajes de error correspondientes se muestran debajo de los respectivos campos de entrada.

Se verifica que los mensajes de error sean claros e informativos.

Se simula la función createQueryArandela para que genere un error.

Se envía el formulario con datos válidos y compruebe si el mensaje de error se muestra en una ubicación destacada.

Se verifica que el mensaje de error esté relacionado con el error generado por la función createQueryArandela.

Se verifica que las selecciones desplegables (tipo de distribución, lleva patín?, línea del producto, tipo de proceso) funcionen correctamente.  
Compruebe que los valores seleccionados se reflejen en los datos del formulario.

### **Componente de registro (Registro):**

Se valida que el componente haga cumplir todos los requisitos del formulario de registro, fortaleza de la contraseña, nombre de usuario único.

Se comprueba que los mensajes de error se muestren correctamente para entradas inválidas.

Se confirma que el componente crea exitosamente una nueva cuenta de usuario y redirige como corresponde.

### **Componente de página de inicio (Home):**

Se verifica que el componente muestre el contenido y los elementos de la interfaz de usuario correctos según el estado de inicio de sesión del usuario.

Se prueba que el componente obtenga y muestre datos de fuentes y/o API relevantes.

Se verifica que los elementos interactivos botones de navegación funcionan como se espera.

### **Pruebas de integración:**

Inicio de sesión y página de inicio:

Se prueba la interacción entre el componente de inicio de sesión y el componente de la página de inicio.

Se verifica que el inicio de sesión exitoso redirija a la página de inicio y actualice la interfaz de usuario en consecuencia.

Se prueba que la página de inicio muestre contenido restringido solo para usuarios registrados.

### **Registro e inicio de sesión:**

Se prueba el flujo entre los componentes de registro e inicio de sesión.  
Verifica que el registro exitoso permita a los usuarios iniciar sesión con sus nuevas credenciales.

Se asegúrate de que el manejo de errores sea consistente en ambos componentes.

### **Pruebas de extremo a extremo:**

Recorrido completo del usuario:

Se simula el registro de un usuario para obtener una cuenta, el inicio de sesión y la navegación a la página de inicio.

Se verifica que todos los componentes interactúen correctamente y que la experiencia de usuario sea fluida.

Se prueba casos excepcionales y posibles problemas que puedan surgir durante las interacciones reales del usuario.

### **Pruebas visuales:**

Consistencia de la interfaz de usuario:

Se utiliza herramientas como Storybook y Jest para garantizar una renderización coherente de los componentes en diferentes navegadores y tamaños de pantalla.

Se verifica que los componentes se adhieran a las pautas de diseño y reglas de estilo definidas.

### **Bibliotecas de prueba:**

Las bibliotecas de prueba usadas fueron React Testing y Jest para agilizar el proceso de prueba.

## **MANTENIMIENTO**

### **PROCEDIMIENTOS DE MANTENIMIENTO:**

#### **Revisiones de Código:**

Realizar revisiones periódicas del código para los componentes y módulos responsables de las funcionalidades principales.

Identificar posibles errores, problemas de rendimiento o ineficiencias.  
Revisar la documentación del código para asegurar que esté actualizada y precisa.

#### **Pruebas Unitarias:**

Ejecutar pruebas unitarias existentes con regularidad para verificar que las funcionalidades continúen funcionando como se espera.

Agregar nuevas pruebas unitarias para cubrir nuevas funcionalidades o cambios en el código.

Actualizar las pruebas unitarias existentes si se realizan cambios significativos en el código.

#### **Pruebas de Integración y E2E:**

Realizar pruebas de integración y de extremo a extremo (E2E) para garantizar que las funcionalidades principales se integren correctamente con el resto de la aplicación.

Realizar estas pruebas después de realizar cambios significativos en el código o en la configuración de la aplicación.

#### **Monitoreo de Errores y Registros:**

Implementar un sistema de monitoreo de errores para capturar y registrar cualquier error o problema que pueda surgir en las funcionalidades principales. Analizar los registros de errores para identificar patrones o tendencias que podrían indicar problemas subyacentes.

Tomar medidas correctivas para solucionar los errores y problemas identificados.

### **Actualizaciones de Dependencias:**

Mantener actualizadas las bibliotecas y dependencias de terceros utilizadas en las funcionalidades principales a sus últimas versiones estables.

Probar cuidadosamente las actualizaciones de dependencias antes de implementarlas en producción.

Considerar posibles cambios en la API o incompatibilidades al actualizar las dependencias.

### **Auditorías de Seguridad y Vulnerabilidades:**

Realizar auditorías de seguridad y evaluaciones de vulnerabilidades con regularidad para las funcionalidades principales.

Identificar y abordar posibles debilidades o vulnerabilidades de seguridad.

Implementar medidas de seguridad para proteger los datos del usuario y prevenir el acceso no autorizado.

### **Optimización del Rendimiento:**

Monitorear el rendimiento de las funcionalidades principales e identificar posibles cuellos de botella o ralentizaciones.

Optimizar el código y las consultas a la base de datos para mejorar el rendimiento.

Implementar mecanismos de caché cuando sea apropiado para reducir la carga en los servidores.

### **Revisiones de Accesibilidad:**

Realizar revisiones de accesibilidad para garantizar que las funcionalidades principales sean accesibles para usuarios con discapacidades.

Implementar las mejoras necesarias para cumplir con los estándares de accesibilidad.

Proporcionar texto alternativo para imágenes y otro contenido no textual.

### **Feedback del Usuario y Análisis de Uso:**



Recopilar feedback del usuario y analizar los datos de uso para identificar áreas de mejora en las funcionalidades principales.

Priorizar las mejoras en función de las necesidades del usuario y los patrones de uso.

Implementar cambios que mejoren la experiencia del usuario y la usabilidad de las características.

### **Documentación:**

Mantener actualizada y precisa la documentación del código y las funcionalidades.

Incluir instrucciones claras sobre cómo usar y mantener las características principales.

Documentar cualquier cambio significativo en el código o las funcionalidades.

## **CONCLUSIONES**

### **RESUMEN DEL PROYECTO:**

Aplicativo de Herramienta Digital para Diseño de Moldes de Inyección de Caucho

**Autor:** Esteban Garcia Zuluaga

**Fecha:** 12/07/2024

**Propietario:** Industrias CADI S.A.S

**Repositorio:** <https://github.com/EstebanGZ-maker/ProyectoCADI>

### **Descripción del Proyecto:**

El proyecto consiste en desarrollar una aplicación digital para el diseño de moldes de inyección de caucho. Esta herramienta está orientada a la digitalización y gestión de datos relacionados con el diseño de moldes, facilitando la interacción entre diseñadores, clientes y usuarios finales. La solución incluye funcionalidades para la búsqueda de productos (O-rings y arandelas) por dimensiones, gestión de archivos, registro y login de usuarios, y centralización de documentación.

### **Objetivos del Proyecto:**

Digitalizar y gestionar la información relacionada con los moldes de inyección de caucho.

Facilitar la interacción tecnológica entre diseñadores, clientes y usuarios.

Centralizar y transformar datos en información útil y accesible.

Crear una herramienta fácil de usar para diseñar moldes de manera eficiente.

### **Componentes del Sistema:**

**Base de Datos:** MongoDB para almacenar productos, usuarios y documentos.

**Back-end:** Desarrollado con JavaScript, node.js, otros frameworks y APIs para la lógica de negocio.

**Front-end:** Interfaces de usuario desarrolladas con HTML, Css React, Typescript y otros frameworks.

### **Funcionalidades:**

- Búsqueda de productos por dimensiones.
- Registro y gestión de usuarios (administradores y usuarios).
- Creación y gestión de productos (O-rings y arandelas).
- Gestión de archivos relacionados con los productos.
- Interfaces de usuario intuitivas para mejorar la experiencia del usuario.

### **Arquitectura del Sistema:**

El sistema se basa en una arquitectura cliente-servidor con un servidor que maneja la lógica de negocio y consultas, y una interfaz de usuario web.

## **RECOMENDACIONES FUTURAS:**

### **Optimización del Rendimiento:**

Implementar técnicas de caché para consultas frecuentes.

Revisar y optimizar consultas a la base de datos para mejorar tiempos de respuesta.

### **Seguridad:**

Realizar auditorías de seguridad periódicas para identificar y corregir vulnerabilidades.

Asegurar que todas las comunicaciones estén encriptadas.

### **Usabilidad:**

Recoger feedback de los usuarios para mejorar la interfaz de usuario.

### **Escalabilidad:**

Preparar la infraestructura para manejar un aumento en la cantidad de datos y usuarios.

Considerar el uso de servicios en la nube para escalar recursos según la demanda.

## **ANEXOS**

### **GLOSARIO DE TÉRMINOS**

**API** (Interfaz de Programación de Aplicaciones): Conjunto de definiciones y protocolos para construir e integrar software de aplicaciones.

**CRUD** (Crear, Leer, Actualizar, Eliminar): Operaciones básicas en la gestión de datos.

**Frontend**: Parte del desarrollo web que se encarga de la interfaz de usuario y la experiencia del usuario.

**Backend**: Parte del desarrollo web que se encarga de la lógica de negocio y la gestión de datos.

**MongoDB**: Base de datos NoSQL que utiliza documentos similares a JSON.

**React**: Biblioteca de JavaScript para construir interfaces de usuario.

**UML** (Lenguaje de Modelado Unificado): Lenguaje de modelado estándar en la ingeniería de software.

**UX/UI** (Experiencia del Usuario/Interfaz de Usuario): Disciplina que se enfoca en mejorar la interacción y experiencia del usuario con el producto.

**DB**: Base de datos.