



**PROYECTO FIN DE CICLO**

**CICLO: DAW**

# **InPrint**

**Fase: 1 – Identificación de necesidades y  
diseño del proyecto**

**Pablo Sánchez Moreno**

*Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.*

*Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.*

*Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.*

*Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.*

*© 2023 Fomento Ocupacional FOC SL todos los derechos reservados.*

## Índice

1 Estudio inicial previo a la realización del proyecto.....	4
1.1. Clasificar las empresas del sector por sus características organizativas y el tipo de producto o servicio que ofrecen.....	4
1.2. Poner un ejemplo de estructura organizativa para una empresa del sector.....	4
1.3. Identificar las necesidades demandas que cubre el proyecto y asociarlas con las necesidades del cliente.....	4
1.4. Descripción del proyecto.....	4
1.5. Justificar el tipo de proyecto elegido para dar solución al problema.....	4
1.6. Características principales del proyecto elegido.....	4
2. Identificación de necesidades y diseño del proyecto.....	5
2.1. Estudio inicial y planificación del proyecto.....	5
2.2. Aspectos fiscales y laborales.....	5
2.3. Viabilidad económica.....	5
2.4. Modelado de la solución.....	6
3. Ejecución del proyecto y pruebas.....	6
3.1. Riesgos de ejecución del proyecto.....	6
3.2. Documentación de ejecución.....	6
3.3. Incidencias.....	7
4. Pruebas y soporte.....	7
4.1. Crear documento con las pruebas a realizar.....	7
4.2. Registro de las pruebas realizadas.....	7
4.3. Evaluar que el proyecto cumple todo lo requerido.....	7

(Una vez realizado el informe, no olvidar actualizar esta tabla del índice, con el fin de que se actualicen todos los epígrafes y números de página)

# Inprint DAW

## 1 Estudio inicial previo a la realización del proyecto

(Dedicación total estimada: 5 horas).  
(Extensión aproximada: 1 a 1 ½ folios máximo)

### 1.1. Clasificar las empresas del sector por sus características organizativas y el tipo de producto o servicio que ofrecen

Aunque existen varias empresas que se dedican en España tanto a la impresión 3D como al mecanizado de piezas, no conozco ninguna que se dedique a ambas. Si bien la mas conocida en impresion 3D en el ambito nacional es Innova 3D existen otras como Imprimakers que ofrecen servicios que se adaptan a los clientes mas amateurs en el sector.

Para el mecanizado se suele tratar de empresas más específicas y que no suelen ir destinadas al público general. En el plano nacional hay algunas como Mecanizados Garrigues, pero a nivel europeo, la alemana Xometry eu tiene un peso mayor.

Algunas de estas empresas te ofrecen el diseñar ellos el concepto que buscas mientras que otras precisan que tu les mandes el diseño ya hecho.

### 1.2. Poner un ejemplo de estructura organizativa para una empresa del sector

En mi opinión la forma más adecuada para empezar una empresa de este tipo consiste en una estructura organizativa basada en ser autónomo.

Esto es debido a que si bien requiere que la persona que se dedica a esto tenga conocimientos en muchos ambitos, cuenta con la ventaja de que la mayor parte de la carga de trabajo es asumida por la maquinaria y no por la persona.

La parte de la aplicación web la realizaría yo mismo ayudado por mis conocimientos y algunos conocidos que podrían ayudarme con temas con los que me costara más desenvolverme

### **1.3. Identificar las necesidades demandas que cubre el proyecto y asociarlas con las necesidades del cliente.**

Con el crecimiento de las tecnologías de impresión 3D ha quedado demostrado que el prototipado de elementos está en pleno auge y su demanda es alta.

Esta tecnología se ha acercado mucho al público general en los últimos años y la gente con menos experiencia a menudo se ven frustradas cuando compran este tipo de máquinas con bajo presupuesto y posteriormente no reciben los resultados que preveen.

Ahí es donde esta empresa apunta a la necesidad del cliente de alcanzar los objetivos que ellos por desconocimiento o por falta de inversión en maquinaria no son capaces de llegar.

### **1.4. Descripción del proyecto.**

El proyecto consiste en crear una web donde el cliente potencial pueda ver a que se dedica la empresa y registrarse o loguearse en ella para enviar los diseños que ellos previamente han realizado o descargado. De este modo desde la parte de la empresa recibirá esos archivos y comenzará con su producción. Durante todo el proceso deberá haber una trazabilidad en cuanto a estado del trabajo se refiere.

### **1.5. Justificar el tipo de proyecto elegido para dar solución al problema.**

A priori la aplicación solo estará disponible en versión de escritorio principalmente debido a que la carga de archivos y el manejo de la aplicación puede hacerse más difícil a los usuarios menos experimentados.

Por otro lado, se presupone que una persona que quiere solicitar el prototipado de un archivo se debe de manejar relativamente bien en escritorio.

## **1.6. Características principales del proyecto elegido.**

Se va a realizar una aplicación web que permite la subida de archivos de objetos 3D que permite a la empresa ficticia recepcionarlos para su posterior fabricación. La empresa a su vez podrá mediante la aplicación dar un seguimiento al cliente de las diferentes fases que atraviesa su pedido.

## 2. Identificación de necesidades y diseño del proyecto

### 2.1. Estudio inicial y planificación del proyecto

- 2.1.1. Identificar las fases del proyecto y su contenido.

Si bien no tengo establecidas unas fases bien diferenciadas, me gustaría seguir un planning aproximado:

- 26 de octubre debería tener completa la parte estructural de la aplicación y sus funcionalidades
- 5 de noviembre debería tener la parte visual de la web completa
- 15 de noviembre debería tener en esos 10 días desde la anterior fase para probarlo todo y terminar de rematar diversos errores que puedan surgir

- 2.1.2. Especificar los objetivos del proyecto.

- Dotar al sitio web de un mecanismo de subida de archivos
- Implementar un servicio de registro y logueo
- Manejar los datos de los usuarios de forma segura mediante encriptación
- Desarrollar un sistema de seguimiento de los pedidos
- Configurar un servidor en local
- Establecer una base de datos y conectarla a la aplicación

- 2.1.3. Especificar recursos hardware y software.

Para la parte de la aplicación:

- La aplicación se va a realizar mediante el uso del entorno de desarrollo Visual Studio Code.
- La parte del frontend se realizará en HTML, CSS y Javascript utilizando React.
- Por su parte, el backend estará programado en Python.
- Para el servidor local utilizaré Uvicorn.
- Para la base de datos utilizaré AWS y para su gestión DBeaver

Para la parte de la empresa:

- Los software que se utilizan en impresión son Autodesk NetFabb y Meshmixer
- Para la parte del mecanizado se usará Ivoclar Cam V3

▪ 2.1.4. Especificar recursos materiales y personales.

En cuanto a recursos materiales la empresa necesitará:

- 3 pc: uno dedicado a la gestión y administración de la empresa, y el otro dedicado a la creación y mantenimiento de la aplicación, y un tercero para la gestión y uso de la maquinaria de la empresa.
- 1 impresora 3d de resina: marca Anycubic, dan buen rendimiento para su precio, en el futuro se podrían agregar más ya que su coste es bajo.
- 1 fresadora 5 ejes: marca VHF, buen rendimiento y precio contenido.
- 1 compresor de aire de al menos 4 bares
- 1 local: no es necesario un gran tamaño, pero si debe tener diferenciadas las siguientes zonas, una sala para la fresadora y su compresor, una sala para la impresora, una sala común donde podrían estar ambos ordenadores.

En cuanto a recursos personales:

- Lo ideal sería llevarla entre dos personas: yo me encargaría de la aplicación y la fabricación de pedidos, y mi pareja se podría encargar de la administración y contabilidad de la empresa.

▪ 2.1.5. Realizar una asociación de fases y recursos materiales que deben intervenir en cada fase

Yo aprecio dos fases diferenciadas:

- En la primera de formación de la empresa y el local, la maquinaria ha de ser colocada en su lugar y configurada correctamente
- En la segunda, cuando la empresa está estabilizada el papel de la maquinaria y los pc es fundamental y de ellos depende el correcto funcionamiento de la empresa

▪ 2.1.6. Realizar una asociación de fases y recursos humanos que deben intervenir en cada fase

En las fases anteriores:

- En la primera fase, la creación de la empresa y la aplicación yo me podría encargar de crear la aplicación con uno de los pc, preparar e instalar la maquinaria y configurarla en el segundo pc y mi pareja se encargaría de contactar con clientes potenciales y promocionar el producto entre sus conocidos y mediante las redes sociales



- En la segunda fase, ya teniendo una cartera de clientes yo podría centrarme más en la maquinaria, mientras que mi pareja se encargaría más de la contabilidad, la mensajería, y la atención al cliente

(Extensión aproximada: 2-3 folios máximo)

## 2.2. Aspectos fiscales y laborales

- Determinar las obligaciones fiscales, laborales y de prevención de riesgos laborales.

### Obligaciones Fiscales:

- Alta en Hacienda: dar de alta en el registro de autónomos
- Declaración de Impuestos: como autónomo, presentar declaraciones trimestrales o anuales de impuestos sobre el valor añadido (IVA) y sobre la renta.
- Facturación y contabilidad: llevar un registro adecuado de ingresos y gastos. Emitir facturas correctamente y mantener una contabilidad ordenada.
- Retenciones IRPF: aplicar retenciones del Impuesto sobre la Renta de las Personas Físicas (IRPF)

### Obligaciones Laborales:

- Seguridad Social: realizar contribuciones a la seguridad social.

### Prevención de Riesgos Laborales:

- Evaluación de riesgos: realizar una evaluación de riesgos laborales para identificar posibles peligros en tu lugar de trabajo.
- Medidas preventivas: implementar medidas preventivas para reducir o eliminar los riesgos identificados.
- Formación: obtener formación en seguridad y salud laboral a nosotros mismos y a cualquier futuro empleado.
- Equipos de Protección Individual (EPI): utilizar EPI adecuados según sea necesario.

Aún así hablaría con un asesor fiscal y un profesional en prevención de riesgos laborales de mi área para obtener información específica y personalizada sobre mis obligaciones.

(Extensión aproximada: 1 folio máximo).

## 2.3. Viabilidad económica

- 2.3.1. Realizar un presupuesto económico del proyecto.

### **Presupuesto Económico:**

#### Recursos Materiales:

- Local de 60 metros cuadrados en Madrid: alrededor de 600 euros mensuales
- 3 Ordenadores de gama media: 3 x 500 euros
- Impresora 3D de gama media: 500 euros
- Fresadora CNC de 5 ejes: 2000 euros
- Herramientas y Equipos Adicionales: 500 euros
- Mobiliario de oficina y equipamiento: estantes, mesas de trabajo, etc.: 1000 euros

#### Licencias y Permisos:

- Licencias comerciales y permisos locales: 1300 euros

Tarjetas de visita, material promocional: 200 euros

Reserva para Gastos Inesperados: 1500 euros

Contingencia (aproximadamente el 10% del presupuesto total).

Total Estimado: alrededor de 10.000 euros

- 2.3.2. Identificar la financiación necesaria.

El dinero provendría al 50% de mis ahorros personales, y el otro 50% de los ahorros de mi pareja. Con eso cubriríamos el proyecto pero si hiciese falta más, sería posible pedir algún tipo de crédito.

■ 2.3.3. Detallar posibles ayudas y subvenciones

En la comunidad de Madrid podríamos percibir una serie de ayudas tales como:

**Programas de Apoyo al Emprendimiento:**

- Ayudas del Programa EMPRENDETUR EMPRESAS: Destinadas a fomentar el emprendimiento en el sector turístico.
- Programa de Promoción del Empleo Autónomo: Incentiva la creación de empleo autónomo.

**Subvenciones para la Innovación y Tecnología:**

- Programa de Cheques de Innovación: Ofrece ayudas para la realización de proyectos de innovación.
- Programa INNVIERTE: Apoyo a proyectos de I+D+i.

**Subvenciones para la Internacionalización:**

- Programa Madrid Exporta: Dirigido a fomentar la internacionalización de las empresas madrileñas.

(Extensión aproximada: de 1 a 1 ½ folios máximo).

## 2.4. Modelo de solución

### ■ 2.4.1. Modelado de la solución:

La aplicación se va a basar en un proyecto de React.

Se dividirá en dos partes:

- ✓ Un frontend, en HTML, CSS y JavaScript para la parte visual de la aplicación en interacción con el cliente
- ✓ Un backend que gestionará la lógica de la aplicación y la conexión de la misma con la base de datos.

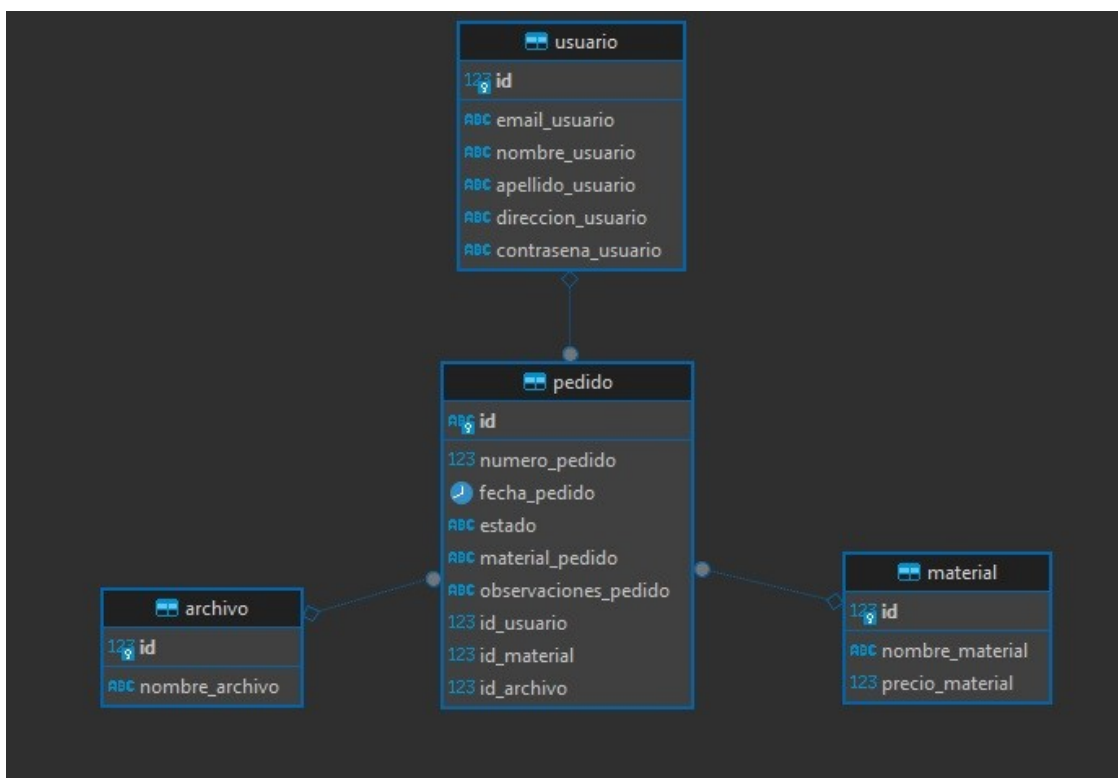
Crearé una base de datos con AWS para aportar mayor complejidad y credibilidad a la aplicación.

La base de datos será PostgreSQL. La razón es por la facilidad que me aportará su conexión con la aplicación y porque es uno de los tipos que Amazon RDS pone a mi servicio.

En esta base habrán una serie de tablas:

- La tabla “pedido”, que contendrá información sobre los pedidos, incluyendo el usuario que realizó el mismo, el material elegido y el archivo asociado al pedido.
- La tabla “usuario” se utilizará para gestionar la información de los usuarios.
- La tabla “material” contendrá información sobre los materiales disponibles.
- La tabla “archivo” que contendrá los detalles sobre los archivos subidos.

Este es el diagrama de entidad-relación que seguirá mi base de datos:



Para la correcta creación de las tablas y gestión en general de la base de datos, a parte de todas las opciones que nos da AWS, utilizaré el software Dbeaver, que utiliza la interfaz de programación de aplicaciones JDBC para interactuar con las bases de datos a través de un controlador JDBC.

La idea es que las contraseñas de los usuarios estén encriptadas para una correcta política de privacidad y para esto utilizaré una librería (Crypto-js) que me permita gestionarlo.

A la aplicación se le va a dotar de la posibilidad de subir archivos, por lo que estos se deben poder manejar correctamente.

En la parte de frontend el proyecto se realiza con React por varios motivos.

Principalmente es que su uso durante las prácticas me ha hecho conocerlo y apreciarlo por su facilidad de uso y porque utiliza JSX (JavaScript XML), una extensión de JavaScript que permite escribir código HTML directamente en archivos JavaScript. Esto mejora la legibilidad y la mantenibilidad del código, además de proporcionar una sintaxis más expresiva.

Además cuenta con extensiones y herramientas de desarrollo que facilitan la depuración y el perfilado de aplicaciones como React DevTools, algo realmente útil.

Para los recursos gráficos hay que tener en cuenta que todas las imágenes que aparezcan deben estar libres de copyright, para ello, nos ayudaremos de páginas de creación de imágenes mediante IA, tales como Midjourney o DALL-E, y de CorelDRAW para la vectorización de imágenes. También me apoyaré en EZGIF para la optimización de imágenes.

En el backend voy a utilizar Python como lenguaje de programación principal.

Para la realización de consultas a la base de datos me ayudaré de la librería SQLAlchemy, esta me proporcionará una forma mucho más sencilla de hacer queries sobre la base de datos ya que no requiere de conocer/usar SQL

Existen otras alternativas en Python como Peewee, y otros lenguajes tienen sus propios ORMs como PHP Eloquent o Java Hibernate, pero me quedo con esta ya que la he utilizado durante mis prácticas en la empresa y me ha fascinado la facilidad de uso que nos brinda.

- 2.4.2. Detalle de los puntos que se van a controlar para validar el proyecto.

Los puntos que considero fundamentales en la aplicación son los siguientes:

- Aspecto visual: este debe de tener una estética agradable y conseguida.
- Seguridad: como es la parte que más me interesa, quiero que los usuarios tengan un registro y un login que pueda ser aplicable a cualquier aplicación real, es decir, que sus credenciales se encripten y se guarden en la base de datos y que yo como administrador de las mismas no sea capaz de averiguar la contraseña de ningún usuario del mismo modo que no lo podría hacer ningún atacante aunque consiguiera hacer una brecha de seguridad y acceder a los registros de la base de datos.
- Que haya una función manejadora de los archivos que se suben a la página, que controle si se sube uno o varios archivos y sobre todo y más importante, que compruebe que el tipo de archivo es correcto y deniegue si es de otro tipo para evitar que se puedan subir a la aplicación cualquier tipo de software o script malicioso.
- El resto de aspectos de la aplicación los considero en un segundo plano, ya que quiero centrarla en ser segura. Todo lo que haya extra es bien recibido pero no prioritario

## 3. Ejecución del proyecto y pruebas

### 3.1. Riesgos de ejecución del proyecto

- 3.1.1. Identificación de riesgos del proyecto.

Estos son algunos de los posibles riesgos asociados:

- Riesgos de salud y seguridad:
  - Lesiones laborales: posibles accidentes durante la operación de maquinaria, como la fresadora CNC, que podrían resultar en lesiones físicas
  - Exposición a sustancias tóxicas: utilizando materiales especiales para la impresión 3D, podría haber riesgos asociados con la exposición a sustancias tóxicas o partículas.
- Riesgos de recursos:
  - Escasez de materiales: problemas en la cadena de suministro que podrían llevar a la escasez de materiales necesarios para la impresión 3D y el mecanizado.
  - Fallas de equipos: posibles problemas técnicos o fallos en los ordenadores, impresora 3D o la fresadora CNC, lo que podría afectar la producción.
- Riesgos de financiación:
  - Variabilidad en los costos de operación: fluctuaciones inesperadas en los costos operativos que podrían afectar la estabilidad financiera del negocio.
- Riesgos con clientes:
  - Incumplimientos de plazos: dificultades para cumplir con los plazos de entrega, lo que podría generar insatisfacción por parte de los clientes.
  - Expectativas no satisfechas: posibles discrepancias entre las expectativas del cliente y el producto final entregado.
- Riesgos legales y regulatorios:
  - Incumplimiento normativo: no cumplir con las normativas locales en materia de seguridad laboral, protección del medio ambiente u otras regulaciones relacionadas con la industria.

Es esencial realizar una evaluación continua de riesgos y ajustar las estrategias de mitigación según sea necesario a lo largo del proyecto.

---

■ 3.1.2. Creación de plan de prevención de riesgos.

**Plan de prevención de riesgos:**

Medidas Preventivas:

- Riesgos de Salud y Seguridad:
  - Lesiones laborales: proporcionar y hacer cumplir el uso de equipos de protección personal (EPI) adecuados, como guantes, gafas de seguridad y calzado de protección. Implementar procedimientos operativos seguros y realizar auditorías regulares para garantizar el cumplimiento
  - Exposición a sustancias tóxicas: utilizar impresoras 3D y materiales que cumplan con las normativas de seguridad y que minimicen la emisión de sustancias tóxicas. Proporcionar formación sobre los riesgos asociados con los materiales utilizados y cómo manejarlos de manera segura.
- Riesgos de recursos:
  - Escasez de materiales: diversificar proveedores y mantener una red de contactos para facilitar la adquisición de materiales alternativos en caso de escasez. Mantener un inventario adecuado para cubrir posibles interrupciones en la cadena de suministro. Establecer contratos a largo plazo con proveedores confiables para asegurar la disponibilidad continua
  - Fallas de equipos: implementar un programa regular de mantenimiento preventivo para todos los equipos, incluyendo ordenadores, impresora 3D y fresadora CNC. Contar con personal capacitado para realizar reparaciones básicas y resolver problemas técnicos de manera eficiente
- Riesgos de financiación:
  - Variabilidad en los costos de operación: mantener registros financieros precisos y realizar un seguimiento regular de los gastos operativos
- Riesgos con clientes:
  - Incumplimientos de plazos: establecer plazos realistas basados en la capacidad de producción y las condiciones del mercado. Comunicarse proactivamente con los clientes en caso de retrasos y buscar soluciones alternativas.
  - Expectativas no satisfechas: realizar revisiones regulares con los clientes para asegurarse de que sus expectativas se estén cumpliendo. Fomentar la comunicación abierta y transparente para abordar cualquier problema o malentendido de manera inmediata.
- Riesgos legales y regulatorios:
  - Incumplimiento normativo: mantenerse actualizado sobre las regulaciones locales, nacionales e internacionales que afectan a la industria de fabricación. Colaborar con profesionales legales y consultores para asegurar el cumplimiento normativo



## 3.2. Documentación de ejecución

- 3.2.1. Indicar las necesidades (si las hubiese) en cuanto a permisos para la puesta en marcha del proyecto (por ejemplo si hay que pedir un permiso de obra), o algún tipo de permiso sobre LOPD, etc....

Estos son los permisos que debería tener en consideración:

- Licencia de actividad comercial
  - Cumplimiento de normativas de seguridad (implementación de medidas específicas para la maquinaria, manipulación de materiales y gestión de residuos)
  - LOPD (Ley Orgánica de Protección de Datos) dado que se maneja información personal de clientes, proveedores y empleados
  - Licencia ambiental debido a que se utilizan materiales o procesos que pueden tener impacto en el medio ambiente
  - Seguro de responsabilidad civil: que cubre posibles daños a terceros
  - Cumplimiento de normativas de propiedad Intelectual: ya que al trabajar con diseños propios o de terceros, hay que asegurarse de respetar los derechos de propiedad intelectual.
- 3.2.2 Ejecución del proyecto.

Aquí voy a explicar un poco el funcionamiento interno de la aplicación.

Empezaré por el frontend. Partimos en Visual Studio de un proyecto que se divide en dos carpetas principales: inprint (frontend) y app (backend). El frontend es un proyecto de Node React, que se compone de una serie de archivos y las librerías colaboran en su funcionamiento.

Para iniciar un proyecto de este tipo primeramente debemos tener instalado Node.js y npm que es el administrador de paquetes de Node. Instalamos React haciendo uso de npm. Creamos el proyecto por comandos de terminal y ya tenemos una base sobre la que trabajar.

Por defecto, el archivo principal que administra la aplicación es Index.js, que nosotros derivamos a Home.js para trabajar todos los componentes desde un mismo directorio.

En React todo se basa en componentes. Estos componentes, valga la redundancia, se componen de tres secciones diferenciadas.

- En la primera sección encontraríamos los imports, estos serían los recursos externos al componente de los que se va a hacer uso en el mismo; un ejemplo podría ser una imagen, u otro componente.

```
JS Home.js  X
inprint > src > components > JS Home.js > Home > showLoginModal
1  import React, { Component } from 'react';
2  import 'bootstrap/dist/css/bootstrap.min.css';
3  import 'bootstrap/dist/js/bootstrap.bundle.min.js';
4  import Navbar from './Navbar';
5  import fondoAvif from './media/fondo.avif';
6  import car1 from './media/impresion1.webp';
7  import car2 from './media/impresion2.webp';
8  import car3 from './media/impresion3.webp';
9  import fondoPNG from './media/fondo.png';
10 import '@fortawesome/fontawesome-free/css/all.css';
11 import LoginModal from './LoginModal';
12 import RegisterModal from './RegisterModal';
13 import UploadFile from './UploadFile';
14 import GeneralView from './GeneralView';
15 import Footer from './Footer';
16
```

*Ejemplo de imports en Home.js*

- La segunda sección la formarían el constructor y las funciones que se utilizan en el componente; en esta sección se encontraría la mayor parte de la lógica del componente, como por ejemplo la función que detecta si el usuario está logueado, o la que verifica si el archivo que se está subiendo es el correcto.

```
async getDeviceType() {
  try {
    let ua = await navigator.userAgent;
    if (/^(tablet|ipad|playbook|silk)|(android(?!. *mobi))/i.test(ua)) {
      return "tablet";
    }
    if (
      /Mobile|iP(hone|od)|Android|BlackBerry|IEMobile|Kindle|Silk-Accelerated|(hpw|web)OS|Opera M(obi|ini)/.test(ua)
    ) {
      return "mobile";
    }
    return "desktop";
  } catch (err) {
    console.error('Error al detectar el tipo de dispositivo', err);
    throw err;
  }
}
```

*Ejemplo de función que detecta el tipo de dispositivo*

- Por último se encuentra el render; esta sección es la que se encarga del aspecto visual del componente, en el caso del Navbar por ejemplo, se define en esta sección qué aspecto debe tener para según que caso nosotros establezcamos. Esto permite que el Navbar siempre sea el mismo componente pero los botones que tiene varíen en función de si por ejemplo el usuario esta logueado o no. No tendría sentido que se mostrase el botón de iniciar sesión cuando ya está iniciada, o que se mostrase un botón de cerrar sesión cuando aún no se ha iniciado.

```
render() {
  // try{

    let relleno

    console.log("token de navbar: ",this.props.userToken)

    //si hay token
    if (this.props.userToken !== "null") {

      //con cerrar sesion
      relleno =
        <table style={{width: "100%", backgroundColor: "#7CF7AB", padding: "10px", borderRadius: "10px" }} >
          <tr style={{ width: "100%" }}>
            <td style={{ width: "20%" }}>
              <div style={{ width: "100%", marginTop: "10px", marginBottom: "10px", marginLeft: "10px" }}>
                <button className="btn btn-light" style={{ width: "150px" }}>Inicio</button>
              </div>
            </td>
          </tr>
        </table>
    }
  }
```

#### *Ejemplo de render en Navbar.js*

Nuestro componente principal es Home. De él dependen el resto que le acompañan en la carpeta (Navbar, Footer, UploadFile...). Este archivo Home se compone de los aspectos básicos, el núcleo de la aplicación. Esta es reactiva, y en función de una serie de condiciones hace que se desplieguen unos u otros componentes. Es decir, si hacemos click en el botón "Registrarse" que se encuentra en el componente Navbar, este sabe que tiene que desplegar el componente RegisterModal; si el usuario cierra la ventana modal, el componente Home reacciona y sabe que tiene que desplegarse en su lugar; y así con todos sus elementos. Esto facilita el mantenimiento del código y evita muchas duplicidades a la hora de trabajar sobre él.

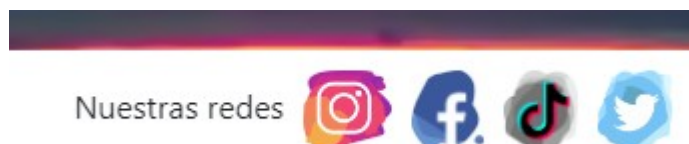
Un componente hijo puede utilizar funciones existentes en un componente padre, pero para ello, se deben establecer desde el componente padre los parámetros y funciones a los que el componente hijo va a poder acceder. Para ello hacemos un bindeo de los mismos.

```
this.onClose = this.onClose.bind(this)
this.onCloseNav = this.onCloseNav.bind(this)
this.onCloseLog = this.onCloseLog.bind(this)
this.onCloseReg = this.onCloseReg.bind(this)
this.showLoginModal = this.showLoginModal.bind(this)
this.showRegisterModal = this.showRegisterModal.bind(this)
this.putLoading = this.putLoading.bind(this);
```

#### *Ejemplo de bindeos desde Home.js*

Estos componentes hijos podrán acceder a estas características del componente padre por medio de props.

En el frontend también tenemos una carpeta específica, llamada Media, que contiene todos los recursos gráficos de la aplicación. Para lograrlos me he ayudado de DALL-E para la imagen de fondo, mientras que para los logotipos de las redes sociales o de la empresa ficticia los he diseñado con CorelDRAW, de este modo evito el copyright y le doy un toque personal.



Por parte del backend, utilizo FastAPI, que es un marco de desarrollo web moderno para crear APIs (Interfaces de Programación de Aplicaciones) rápidas con Python 3.7+ basado en estándares abiertos y tecnologías estándar de la industria.

Con FastAPI lo que hacemos es tener una serie de endpoints a los que accedemos por medio de requests. El backend a su vez está conectado a la base de datos mencionada con anterioridad por medio de SQLAlchemy, que para gente inexperta como yo facilita los queries de SQL.

```
from sqlalchemy.orm import declarative_base, sessionmaker
from sqlalchemy import create_engine, engine
from decouple import config

try:
    SQLALCHEMY_DATABASE_URL = config('SQLALCHEMY_DATABASE_URL')
except:
    SQLALCHEMY_DATABASE_URL = "postgresql://postgres:Sa17031997!@inprintdb.c9t
engine = create_engine(SQLALCHEMY_DATABASE_URL)
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)

Base = declarative_base()
```

*Archivo database.py que configura la conexión a la base de datos*

En la aplicación es muy importante que todos los datos que tratamos sigan el mismo modelo que en la base de datos, para ello utilizamos modelos y esquemas. Estos modelos son la representación de las tablas, columnas, tipos de datos y relaciones de las tablas alojadas en nuestra db. Nosotros los almacenamos como objetos.

```
class Archivo(Base):
    __tablename__ = 'archivo'
    id = Column(Integer, primary_key=True)
    nombre_archivo = Column(String, nullable=False)

class Material(Base):
    __tablename__ = 'material'
    id = Column(Integer, primary_key=True)
    nombre_material = Column(String, nullable=False)
    precio_material = Column(Integer)

class Usuario(Base):
    __tablename__ = 'usuario'
    id = Column(Integer, primary_key=True)
    email_usuario = Column(String, nullable=False)
    nombre_usuario = Column(String)
    apellido_usuario = Column(String)
    direccion_usuario = Column(String)
    contrasena_usuario = Column(String, nullable=False)
```

*Archivo models.py*

Al igual que en frontend nuestro archivo principal era Home.js, en backend lo es main.py. Aquí es donde se alojan los endpoint de la aplicación. Estos serían como las funciones principales de la misma. Estos realizan consultas y operaciones sobre la base de datos en función de los parámetros que le lanza el frontend mediante los request. Aquí al igual que en frontend las funciones se pueden llamar desde archivos distintos, pero tienen que ser importadas. Un ejemplo sería el obtener un usuario, que a su vez llama a la función `get_current_user` que está alojada en el archivo `global_utils.py`.

```
@app.get("/api/users/me", response_model=schemas.User)
async def get_user(user: schemas.User = Depends(get_current_user)):
    # Devuelve el usuario
    try:
        return user
    except:
        raise HTTPException(status_code=404, detail="We couldn't find this user")
```

*Endpoint /api/users/me alojado en main.py*

La parte del backend donde he hecho más énfasis es la creación de usuarios y logeo de los mismos. Para ello he utilizado JWT y crypto.js. Lo que me permiten hacer estas librerías es hacer una doble encriptación sobre las contraseñas de los usuarios, lo que las vuelve muy seguras frente ataques. Esto se hace mediante la aplicación de sales. Estas sales encriptan las contraseñas de tal modo que ni el administrador de la base de datos, en este caso yo, puedo saber cuál es la contraseña de cada usuario.

```
// Librería para cifrar contraseñas
const CryptoJS = require("crypto-js");
const salt = "farm1990M00";
const salt1 = "f1nd1ngn3m0";
const hashPassword = CryptoJS.SHA256(salt1 + password + salt).toString();
```

### Encriptación desde el frontend

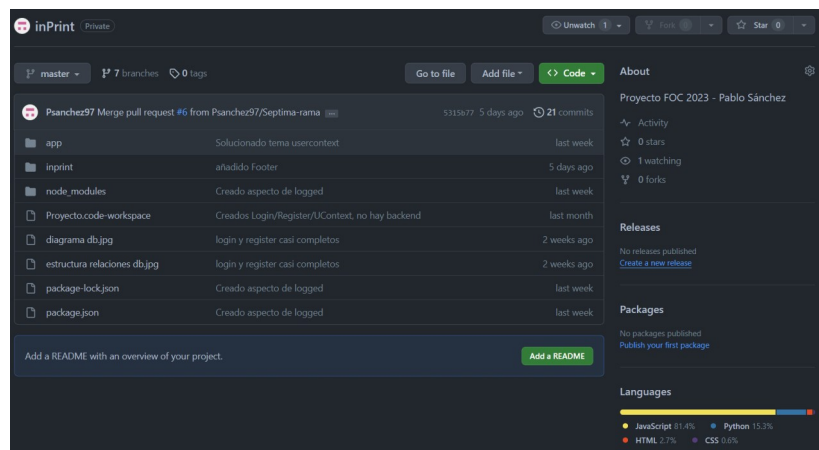
Esas contraseñas que ya desde el frontend se cifran, mediante el uso de JWT se devuelve un token que es manejado desde el frontend de tal modo que trabaja como un identificador único de cada usuario logueado. Esto permitiría que cada usuario vea sus pedidos y no los del resto de clientes, que modifique sus datos y no los de otro, etc...las posibilidades en este aspecto son infinitas.

Key	Value
UserToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbF91c2Vhc...

### Token recibido por el usuario tras logearse

Por motivos de tiempo, cuando un usuario se registra o se loguea, se muestran unos pedidos de ejemplo ya que no he podido llegar a implementar el control de los pedidos de cada usuario.

Como observación final me gustaría añadir que todo el proyecto ha sido llevado mediante un control de versiones utilizando GitHub. Esto me ha permitido llevar un mejor control del desarrollo y lo he utilizado como backup en caso de que ocurriese algo con mi pc.



- 3.2.3. Manuales finales:
  - Manuales de usuario

## Manual de Usuario de Inprint

### 1. Introducción

#### 1.1 Objetivo del Manual

Este manual proporciona información detallada sobre cómo utilizar la aplicación Inprint

#### 1.2 Descripción General de la Aplicación

Inprint es una plataforma que facilita el proceso de realización de pedidos de fabricación personalizada. Los usuarios pueden iniciar sesión, registrar nuevas cuentas y subir archivos en formato .stl para realizar pedidos en los que han de seleccionar opciones de fabricación y material.

#### 1.3 Requisitos del Sistema

- Navegador web actualizado (se recomienda Google Chrome, Mozilla Firefox, o Safari).
- Conexión a Internet estable.

### 2. Registro e Inicio de Sesión





## 2.1 Crear una Cuenta

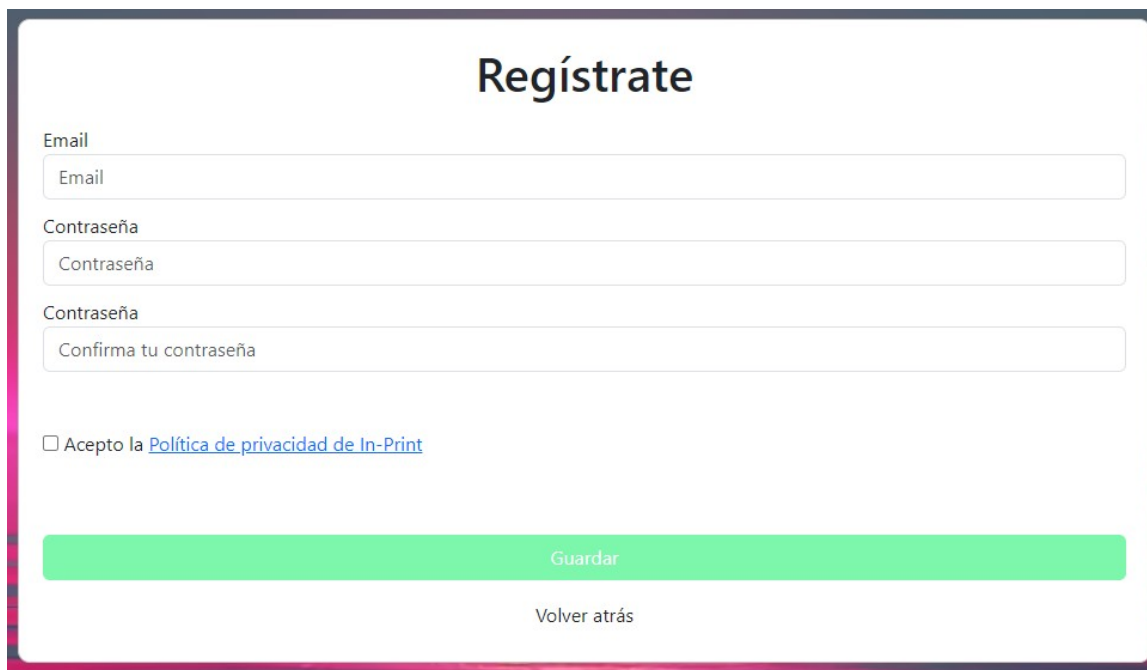
Accede a la página de inicio.

Haz clic en "Registrarse".

Completa el formulario con la información requerida, email, contraseña y confirmación de contraseña (mínimo 8 caracteres).

Acepta la política de privacidad.

Haz clic en "Guardar".



**Regístrate**

Email

Email

Contraseña

Contraseña

Contraseña

Confirma tu contraseña

☐ Acepto la [Política de privacidad de In-Print](#)

Guardar

Volver atrás

## 2.2 Iniciar Sesión

Accede a la página de inicio.

Ingresa tu correo electrónico y contraseña.

Haz clic en "Entrar".



**Inicia sesión**

Email

Email

Contraseña

Contraseña

Entrar

Volver atrás



### 3. Interfaz de Usuario

#### 3.1 Navegación

La aplicación consta de un menú a la izquierda y un listado de pedidos a la derecha. En la parte superior se aprecia la barra de navegación.

The screenshot displays the InPrint application interface. At the top right, there is a navigation bar with buttons: Inicio, Sobre nosotros, Más acciones, and Cerrar sesión. On the left, a sidebar titled 'Sube aquí tus archivos STL' contains a form with a dropdown for 'Tipo de Fabricación' (labeled 'Selecciona'), a date input for 'Fecha de entrega' (labeled 'dd/mm/aaaa'), and two buttons: 'Seleccionar archivo' and 'Subir archivo'. The main area on the right is titled 'Listado de Pedidos' and contains a table with the following data:

Fecha de Pedido	Fecha de Entrega	Estado	Nombre de Archivo	Tipo de Fabricación	Opción
2023-11-11	2023-11-15	En proceso	archivo4.stl	Mecanizado	5 ejes
2023-11-12	2023-11-18	Enviado	archivo5.stl	Impresión	Grey V4
2023-11-12	2023-11-20	Terminado	archivo3.stl	Mecanizado	3 ejes
2023-11-13	2023-11-20	Terminado	archivo7.stl	Impresión	Grey V4
2023-11-13	2023-11-18	Enviado	archivo2.stl	Impresión	Transparent Green
2023-11-13	2023-11-20	En proceso	archivo1.stl	Impresión	Grey V4
2023-11-14	2023-11-19	En proceso	archivo6.stl	Mecanizado	3 ejes

At the bottom of the interface, there is a section for 'Nuestras redes' with social media icons for Instagram, Facebook, and Twitter.

#### 3.2 Barra Lateral

En la barra lateral, selecciona el tipo de fabricación, el material deseado y la fecha para la que se solicita (debe ser de al menos 3 días contando el día actual). Bajo estos desplegados se encuentra "Seleccionar Archivo". Al hacer click sobre el se abre un selector de archivos con el gestor de archivos por defecto en tu sistema. Selecciona archivos .stl desde tu dispositivo. Confirma la carga del archivo haciendo click en "Subir archivo"

This is a close-up of the sidebar form titled 'Sube aquí tus archivos STL'. It includes a dropdown menu for 'Tipo de Fabricación' with the placeholder text 'Selecciona'. Below it is a date input field for 'Fecha de entrega' with the placeholder text 'dd/mm/aaaa' and a calendar icon. At the bottom of the form are two green buttons: 'Seleccionar archivo' and 'Subir archivo'.

#### 4. Listado de Pedidos

A la derecha, visualizarás un listado de pedidos realizados.

Información mostrada: fecha de pedido, fecha de entrega, estado, nombre de archivo, tipo de fabricación y material. Hacer click sobre los encabezados de la tabla ordenará los pedidos

Listado de Pedidos					
Fecha de Pedido 1	Fecha de Entrega	Estado	Nombre de Archivo	Tipo de Fabricación	Opción
2023-11-11	2023-11-15	En proceso	archivo4.stl	Mecanizado	5 ejes
2023-11-12	2023-11-18	Enviado	archivo5.stl	Impresión	Grey V4
2023-11-12	2023-11-20	Terminado	archivo3.stl	Mecanizado	3 ejes
2023-11-13	2023-11-20	Terminado	archivo7.stl	Impresión	Grey V4
2023-11-13	2023-11-18	Enviado	archivo2.stl	Impresión	Tranparent Green
2023-11-13	2023-11-20	En proceso	archivo1.stl	Impresión	Grey V4
2023-11-14	2023-11-19	En proceso	archivo6.stl	Mecanizado	3 ejes

#### 5. Opciones de contacto

En la parte inferior hay un acceso directo a todas las redes sociales desde las que se puede contactar con la empresa para cualquier tipo de duda o sugerencia.



#### 6. Acerca de

##### 6.1 Versiones y Fechas de Lanzamiento

Se trata de una la versión 0.8. Aún es un prototipo y no es funcional al 100%

---

- Manuales de instalación

Requisitos previos:

- Visual Studio Code
- Node.js
- Python
- Git

## Pasos para la instalación:

## 1. Clonar el repositorio:

```
git clone https://github.com/Psanchez97/inPrint.git
```

## 2. Instalar dependencias del frontend:

```
cd inprint  
npm install
```

## 3. Instalar dependencias del backend:

```
cd ../app  
pip install -r requirements.txt
```

## 4. Configurar variables de entorno:

En la carpeta app, crea un archivo llamado .env y configura las variables de entorno necesarias (la conexión a la base de datos y la clave JWT).

(en este caso yo ya lo proporciono creado pero sería recomendable que no estuviera en GitHub por contener información sensible)

## 5. Ejecutar el servidor local:

En la carpeta app, ejecuta el siguiente comando para iniciar el servidor:  
uvicorn main:app --reload  
Esto iniciará el servidor FastAPI en localhost:8000

## 6. Ejecutar el frontend:

En la carpeta inprint, ejecuta los siguientes comandos:

```
npm run build  
serve -s build
```

Esto iniciará la aplicación React y abrirá automáticamente tu navegador en <http://localhost:3000>.

¡Listo! Ahora deberías tener la aplicación web ejecutándose en tu entorno local.

### 3.3. Incidencias

#### 3.3.1. Definir un protocolo para resolución de incidencias:

- ✓ 3.3.1.1. Recopilación de información
- ✓ 3.3.1.2. Posible solución
- ✓ 3.3.1.3. Registro

#### Protocolo de Resolución de Incidencias

##### 1. Recopilación de Información

###### 1.1 Descripción del Problema

El usuario debe proporcionar una descripción detallada del problema/incidencia, incluyendo cualquier mensaje de error, pasos previos y condiciones específicas que llevaron al problema.

###### 1.2 Información del Usuario

Recopilar detalles del usuario afectado dirección de correo electrónico.

###### 1.3 Contexto del Sistema

Obtener información sobre el entorno del sistema en el que se produjo la incidencia, incluyendo el dispositivo utilizado, el navegador y cualquier configuración específica.

##### 2. Posible Solución

###### 2.1 Análisis de la Incidencia

La persona especialista en soporte técnico (en esta casuística yo mismo) analizará la información recopilada para comprender la naturaleza y el alcance del problema.

###### 2.2 Identificación de la Causa Raíz

Se realizarán pruebas adicionales, si es necesario, para identificar la causa subyacente del problema.

###### 2.3 Propuesta de Solución

Se propondrá una solución basada en la información recopilada y el análisis realizado.

###### 2.4 Comunicación con el Usuario

El usuario afectado será informado sobre la propuesta de solución junto con instrucciones claras sobre cómo implementarla.

###### 2.5 Confirmación de la Solución

Se verificará con el usuario que la solución propuesta haya resuelto la incidencia de manera satisfactoria.

##### 3. Registro

###### 3.1 Documentación de la Incidencia

Todas las etapas del proceso de resolución se registrarán en un sistema de seguimiento de incidencias. Inicialmente puede ser una hoja de Excel alojada en un repositorio seguro.

###### 3.2 Datos del Usuario

Se registrará la información del usuario afectado, así como cualquier dato relevante relacionado con la incidencia.

3.3 Descripción de la Solución

Se documentará detalladamente la solución implementada, incluyendo cualquier ajuste realizado en el sistema.

3.4 Fecha y Hora

Se registrará la fecha y la hora de inicio y finalización de la resolución de la incidencia.

3.5 Feedback del Usuario

Se recopilará cualquier feedback adicional del usuario sobre la experiencia de resolución de la incidencia.

4. Cierre de la Incidencia

4.1 Confirmación del Usuario

El usuario afectado confirmará la resolución exitosa de la incidencia.

4.2 Seguimiento Post-Cierre

Se realizará un seguimiento posterior al cierre para asegurarse de que la incidencia permanezca resuelta y para abordar cualquier problema adicional que pueda surgir.

Este protocolo garantiza una respuesta estructurada y eficiente ante incidencias, asegurando la satisfacción del usuario y proporcionando datos valiosos para mejorar continuamente la aplicación.

## 4. Pruebas y soporte

### 4.1. Crear documento con las pruebas a realizar

#### Informe de Prueba de Seguridad: Intento de Inicio de Sesión

##### Resumen

La prueba de seguridad se llevó a cabo para evaluar la resistencia del sistema ante intentos de inicio de sesión sin credenciales correctas. El objetivo es identificar posibles vulnerabilidades y garantizar la integridad y seguridad del sistema.

##### Procedimiento

###### 1. Objetivo

Evaluar la capacidad del sistema para detectar y responder a intentos de inicio de sesión no autorizados.

###### 2. Escenario de Prueba

Se simularon múltiples intentos de inicio de sesión con credenciales incorrectas utilizando un conjunto de usuarios ficticios.

##### Resultados

###### 3. Detección de Intentos Maliciosos

El sistema demostró una respuesta adecuada ante los intentos fallidos de inicio de sesión.

Se verificó que los usuarios no pudieran acceder al sistema después del intento.

###### 4. Mensajes de Error

Los mensajes de error generados por el sistema durante los intentos fallidos de inicio de sesión no revelan información sensible.

Los mensajes son genéricos y no proporcionan pistas sobre la validez de las credenciales.

###### 5. Mejoras Sugeridas

Se sugiere realizar una evaluación continua de las políticas de bloqueo de cuentas en función de las amenazas emergentes.

Considerar la implementación de medidas adicionales, como el bloqueo de cuenta ante un número elevado de intentos fallidos de inicio de sesión.

## Conclusiones

La prueba de seguridad demostró que el sistema responde eficientemente ante intentos de inicio de sesión sin credenciales correctas. Las medidas implementadas fortalecen la seguridad del sistema.

## Documento de Prueba de Copias de Seguridad en AWS

### Objetivo

El objetivo de este documento es proporcionar una guía detallada sobre la realización de pruebas de copias de seguridad en la base de datos alojada en AWS. Estas pruebas aseguran la integridad y disponibilidad de los datos críticos almacenados en la base de datos, así como la capacidad para restaurarlos en caso de una pérdida de datos o una interrupción del servicio.

### Procedimiento de Prueba

#### 1. Acceso a la Consola de AWS

Inicia sesión en la Consola de AWS.

Selecciona el servicio "Amazon RDS" en el panel de servicios.

#### 2. Identificación de la Instancia de la Base de Datos

En el panel de Amazon RDS, selecciona la instancia de la base de datos correspondiente.

Verifica que la instancia esté en un estado "Available" y que no haya problemas de salud.

#### 3. Configuración de Copias de Seguridad Automáticas

En la pestaña "Instancias de base de datos" de la consola de Amazon RDS, selecciona la instancia de la base de datos.

Navega a la pestaña "Configuración".

Asegúrate de que las copias de seguridad automáticas estén habilitadas.

Verifica la frecuencia y el período de retención de las copias de seguridad automáticas.

#### 4. Inicio de una Copia de Seguridad Manual

Desde la pestaña "Instancias de base de datos", selecciona la instancia.

Navega a la pestaña "Copias de seguridad" y haz clic en "Crear copia de seguridad ahora".

Confirma que la copia de seguridad se ha iniciado correctamente.

#### 5. Monitorización del Progreso de la Copia de Seguridad

En la pestaña "Copias de seguridad", observa el estado de la copia de seguridad manual.

Asegúrate de que la copia de seguridad se completa sin errores.

#### 6. Restauración de una Copia de Seguridad

Desde la pestaña "Copias de seguridad", selecciona la copia de seguridad que acabas de crear.

Haz clic en "Restaurar a la instancia".

Confirma que la restauración se ha completado correctamente.

#### 7. Verificación de la Integridad de los Datos

Accede a la aplicación y realiza consultas en la base de datos restaurada.

Verifica que los datos restaurados sean coherentes y no haya pérdida de información.

#### Registro de la Prueba

Fecha y Hora de la Prueba: 18/11/2023

Copia de Seguridad Automática:

Estado: Éxito

Copia de Seguridad Manual:

Estado: Éxito

Restauración de Copia de Seguridad:

Estado: Éxito

Verificación de Datos Restaurados:

Estado: Éxito

#### Conclusiones

Todos los procesos se han completado de forma satisfactoria

Este documento debe actualizarse y repetirse periódicamente para garantizar la efectividad continua del proceso de copias de seguridad en AWS.

## 4.2. Registro de las pruebas realizadas

Se han realizado dos pruebas, de seguridad de acceso y de copias de seguridad de la base de datos. Creo que ambas son afines al proyecto y han sido completadas de forma satisfactoria.



### 4.3. Evaluar que el proyecto cumple todo lo requerido.

Objetivos	% Conseguido	Observaciones
Dotar al sitio web de un mecanismo de subida de archivos	80%	El sistema de subida está implementado, pero el posterior tratamiento de esos archivos en la base de datos no esta maneja debido a falta de tiempo
Implementar un servicio de registro y logueo	100%	Conseguido
Manejar los datos de los usuarios de forma segura mediante encriptación	100%	Conseguido y mejorado frente a expectativas iniciales, gracias a dotarle de doble encriptación
Desarrollar un sistema de seguimiento de los pedidos	70%	Implementada la parte visual de este punto, no así la parte lógica, que queda incompleta
Desarrollar un sistema de seguimiento de los pedidos	100%	Conseguido
Establecer una base de datos y conectarla a la aplicación	100%	Conseguido

Revisar punto a punto lo indicado en el punto 2.2. y comprobar que el proyecto lo cumple todo.

(Extensión aproximada: Depende del proyecto. No más de 5 folios en ningún caso).