



UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL SANTA FE

PROYECTO FINAL DE CARRERA

“Sistema de gestión de impresiones para centro de copiado”

Raviolo, Josefina - LU 24044 - josefina.raviolo@gmail.com

Vola, Alejandro Bruno - LU 24050 - aleeevola@gmail.com

Director: Ing Dominguez, Martín

ÍNDICE

1. Introducción	4
1.1 Objetivo general	5
1.2 Objetivos específicos	6
1.3 Alcance	6
2. Marco teórico	7
2.1 Arquitectura	7
2.1.1 Microservicios	7
2.1.2 Monolítica	8
2.2 API Rest	8
2.3 Aplicaciones	9
2.3.1 Aplicaciones de escritorio	9
2.3.2 Aplicaciones web	10
2.3.3 Aplicaciones nativas	11
2.3.4 Conclusión	11
2.4 Cloud Computing	12
3. Dominio	13
4. Herramientas utilizadas	14
4.1 Base de datos	14
4.2 Backend	15
4.3 Frontend	17
4.4 Procesador de pagos	18
4.4 Otras herramientas utilizadas en el desarrollo	19
4.4.1 Integración continua	19
4.4.2 Autenticación	20
4.4.3 Host	20
4.4.4 Swagger	21
4.5 Programas complementarios	22
4.5.1 IDE de Java	22
4.5.2 Editor de código fuente	23
4.5.3 Suite ofimática	24
4.5.4 Microsoft Teams	25
5. Desarrollo	25
5.1 Requerimientos	25
5.1.1 Lista de requerimientos	26
5.1.2 Alcance del proyecto	30

5.2 Metodología	32
5.2.1 Sprint 1	33
5.2.2 Sprint 2	34
5.2.3 Sprint 3	36
5.2.4 Sprint 4	37
5.2.5 Sprint 5	38
5.3 Riesgos	38
5.3.1 Riesgos ocurridos	45
6. Producto	46
6.1 Arquitectura	47
6.2 Modelo de datos	48
6.3 Estructura backend	50
6.4 Funcionalidades	52
Funcionalidades para los clientes	53
6.4.1 Realizar pedido	55
6.4.1.1 Subir archivo	57
6.4.1.2 Seleccionar archivo del repositorio	58
6.4.2 Programar pedido	64
6.4.3 Pagar pedido	67
6.4.4 Buscar pedido	68
Funcionalidades para los administradores	69
6.4.4 Inicio y cierre de sesión	69
6.4.5 Actualizar precios	71
6.4.6 Ver listado de pedidos	72
6.4.6.1 Ver pedidos según estado	73
6.4.7 Ver detalle de pedido	74
6.4.8 Imprimir archivo	75
6.4.9 Cambiar estado del pedido	76
6.4.10 Lista archivos frecuentes	77
6.4.11 Subir archivo frecuente	78
7. Aseguramiento de la calidad	80
7.1 Backend	80
7.1.1 Pruebas automatizadas	80
7.1.1.1 Archivos Frecuentes	81
7.1.1.2 Archivos	81
7.1.1.4 File Storage	81
7.1.1.5 Pagos	81

7.1.1.6 Pedido	82
7.1.1.7 Precio	82
7.1.2 Pruebas API Rest	83
7.2 Frontend	85
7.2.1 Validación del cliente	85
8. Extensibilidad	86
8.1 Refactorización de clases de archivos	86
8.2 Crear cuenta usuarios clientes	86
8.3 Estadísticas de administración	86
8.4 Algoritmos de ordenamiento de tareas	86
8.5 Categorías de archivos frecuentes	87
8.6 Crear cuenta usuarios clientes con diferentes roles	87
8.7 Parametrizar los horarios	87
8.8 Crear un sistema genérico	87
9. Aportes y conclusiones	88
9.1 Desde el punto de vista de la empresa	88
9.1.1 Para la empresa	88
9.1.2 Para los clientes de la empresa	89
9.2 Desde el punto de vista del producto	89
9.3 Desde el punto de vista profesional	90
10. Referencias	92
ANEXO A - Historias de usuario	93

1. Introducción

El presente proyecto consistió en el desarrollo de un sistema web para la gestión de pedidos en un centro de copiado local.

La idea del mismo nace a partir de la observación de la fotocopiadora de la facultad, así como de otros centros de impresión de alta demanda y el flujo de trabajo que realizaban para recibir pedidos online y gestionarlos. Se pudo notar que este proceso consistía en:

1. Enviar un mail a la fotocopiadora.
2. Ir a la fotocopiadora y esperar a ser atendido.
3. Solicitar la impresión del archivo enviado.
4. Pagar.

Este procedimiento era ineficiente, tanto para los clientes que debían esperar para la impresión de su pedido, como para la librería que podía perder clientes y no aprovechaba tiempos muertos.

Originalmente se pretendía hacer un producto genérico, lo que requiere tener más conocimiento en el negocio y poder recorrer los distintos centros de copiado para relevar adecuadamente los distintos procesos de negocio y necesidades de cada uno.

Debido a la situación de pandemia, las universidades y escuelas permanecieron cerradas durante el inicio del presente proyecto, imposibilitando la toma de requerimientos, ya que los centros de copiado de las facultades no operaban de forma normal ni se tenía certezas sobre su continuidad.

Es por esto que se decidió desarrollar una solución para un cliente en particular, logrando así facilitar las reuniones y la captura de requerimientos. El cliente es el centro de impresiones “TOTAL IMPRESIÓN DIGITAL”, y el mismo está ubicado en la ciudad de Santo Tomé.

En el presente informe se utilizarán indistintamente los términos “fotocopiadora” y “centro de copiado” para referirse a dicha empresa.

1.1 Objetivo general

El objetivo general del presente Proyecto Final de Carrera es diseñar y desarrollar un producto de software que permita la gestión de archivos en una imprenta. A su vez, agilice el proceso de gestión de una orden de impresión, desde el envío del documento hasta su pago, dando soporte a la interacción con el cliente mediante una aplicación web, reduciendo el tiempo de espera y optimizando la utilización de recursos de almacenamiento.

1.2 Objetivos específicos

- Desarrollar una interfaz de presentación del sitio web, con información pertinente de la empresa y un paso a paso de cómo realizar un pedido.
- Desarrollar una funcionalidad para el personal del centro de copiado, donde se indiquen los pedidos de impresión con sus correspondientes documentos a imprimir, con nombre de cliente, fecha y hora límite de entrega y observaciones. A su vez diferenciarlos por cada uno de sus estados, como ser: pedidos en proceso de impresión y finalizados esperando retiro.
- Desarrollar una funcionalidad para realizar un pedido, donde los clientes puedan cargar sus archivos, formato, horarios de solicitud, precio, etc. A su vez, gestionar el seguimiento del pedido, donde se indique su estado (creado, pendiente, impreso, cancelado, entregado).
- Implementar un gestor de alertas por correo electrónico que notifique a un cliente el estado de su pedido.
- Implementar un sistema de pagos con tarjeta de crédito/débito (mercado pago).
- Implementar una base de archivos frecuentes para poder elegir y encargar el archivo sin tener que enviarlo.

1.3 Alcance

El alcance de este proyecto incluye el diseño y el desarrollo del sistema web propuesto. En el mismo los clientes podrán generar un pedido, subiendo sus propios archivos o eligiendo del repositorio, especificar las características de impresión de cada uno, programar el horario de retiro (con fecha y hora límite) y pagar online mediante la plataforma Mercado Pago.

A la vez, los usuarios administradores, luego de autenticarse, podrán ver los pedidos, actualizar su estado, imprimir los archivos y subir o eliminar archivos del repositorio.

Así se disponen de dos roles dentro del sistema: cliente y administrador. Para el primer rol no se necesita autenticarse mientras que para el segundo sí es necesario iniciar sesión.

Por su parte, el sistema enviará emails informativos cada vez que los pedidos cambien de estado, y se dará la posibilidad de trackear el estado del pedido.

2. Marco teórico

En este apartado se nombran y se explican brevemente los fundamentos teóricos de los distintos enfoques aplicados en el proyecto “Sistema de gestión de impresiones para centro de copiado”.

2.1 Arquitectura

2.1.1 Microservicios

Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde el sistema se compone por pequeños servicios independientes que se comunican a través de API ligeras y bien definidas. Cada uno de éstos se encuentran alojados en equipos pequeños e independientes.

Cada servicio componente en una arquitectura de microservicios se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios. Estos no necesitan compartir ninguno de sus códigos o implementaciones con otros, cada uno se caracteriza por su autonomía y especificidad.

El principal beneficio es la agilidad, ya que cada uno es específico en su tarea y puede resolverla de manera sencilla sin interacción de los demás. A su vez, aportan libertad tecnológica al proyecto, debido a que no es necesario que cada una de las partes utilicen el mismo lenguaje o configuración de equipo permitiendo elegir la mejor herramienta para cada tarea.

2.1.2 Monolítica

En las arquitecturas monolíticas todos los procesos están estrechamente asociados y se ejecutan como un solo servicio. Esto significa que si un proceso experimenta un pico de demanda, escalará a toda la arquitectura causando un cuello de botella en los recursos.

Agregar o mejorar las características de una aplicación monolítica se vuelve más complejo a medida que crece el código. Esta complejidad limita la experimentación y dificulta la implementación de nuevas ideas.

Este tipo de arquitectura aumenta el riesgo de disponibilidad de la aplicación, porque muchos procesos dependientes y estrechamente vinculados aumentan el impacto del error de un proceso.

2.2 API Rest

Para introducir el concepto de API Rest, primero se explicarán brevemente los conceptos de API y REST.

Una API (interfaz de programación de aplicaciones) es un conjunto de definiciones y protocolos que permiten la comunicación de datos entre aplicaciones. Para eso, la API utiliza peticiones HTTP, que son las responsables de las operaciones

básicas necesarias para la manipulación de datos. Las principales solicitudes HTTP son: POST, GET, DELETE y PUT.

REST, por su parte, es la abreviación de Representational State Transfer, el mismo es un conjunto de restricciones que se utilizan para que las solicitudes HTTP cumplan con las directrices definidas en la arquitectura. Las principales restricciones determinadas por la arquitectura Rest son:

- Cliente-servidor: las aplicaciones existentes en el servidor y el cliente deben estar separadas.
- Sin estado: las requisiciones se realizan de forma independiente, es decir, cada una ejecuta sólo una determinada acción.
- Caché: la API debe utilizar la caché para evitar llamadas recurrentes al servidor.
- Interfaz uniforme: agrupa otros cuatro conceptos en los que se determina que los recursos deben ser identificados, la manipulación de los recursos debe ser a través de la representación, con mensajes autodescriptivos y utilizando enlaces para navegar por la aplicación.

Entonces, cuando se habla de API Rest, significa utilizar una API para acceder a aplicaciones back-end, de manera que esa comunicación se realice conforme a los estándares definidos por el estilo de arquitectura Rest.

2.3 Aplicaciones

2.3.1 Aplicaciones de escritorio

Una aplicación de escritorio es un sistema que se almacena en la computadora del usuario final. La principal ventaja es la ejecución sin necesidad de conexión a internet y que poseen un tiempo de respuesta más rápido.

Hoy en día son las menos utilizadas, debido a que requiere una instalación en el ordenador del cliente, causando dificultad para mantenerlas actualizadas en todos los

equipos y su desarrollo se vuelve dependiente del sistema operativo que se vaya a utilizar.

2.3.2 Aplicaciones web

Una aplicación web es un programa de aplicación que se almacena en un servidor remoto y se accede a través de un navegador, como Google Chrome, Microsoft Edge o Firefox.

Una aplicación web estática es aquella que no cambia cuando un usuario la solicita: el servidor Web envía la página al navegador Web solicitante sin modificarla. Por el contrario, una aplicación web dinámica es aquella en la que el servidor modifica las páginas Web antes de enviarlas al navegador solicitante dependiendo la interacción con el usuario. La naturaleza cambiante de este tipo de páginas es la que le da el nombre de dinámica.

Las aplicaciones web pueden ser diseñadas para una amplia variedad de usos, y pueden ser utilizadas por cualquiera, desde personas individuales hasta organizaciones.

Este tipo de aplicaciones están íntimamente relacionadas con el almacenamiento de datos en la nube, ya que toda la información se guarda de forma permanente en servidores web, los cuales además de alojar dicha información, la envían a los dispositivos móviles o equipos informáticos en cada momento que sea requerida, realizando copias temporales de estos envíos dentro de los equipos y dispositivos que se utilicen. Las empresas que alquilan estos espacios en los servidores web son conocidas como empresas o servicios de Hosting.

Las principales ventajas de las aplicaciones web son:

- No necesitan ser instaladas.
- Se pueden acceder desde distintos dispositivos (computadoras, smartphones, tablets).
- Se pueden acceder desde distintos navegadores.

- Permiten el acceso de múltiples usuarios a la misma versión de la aplicación.
- Accesibles desde cualquier lugar con conexión a Internet.

2.3.3 Aplicaciones nativas

Las aplicaciones nativas son aquellas que se crean específicamente para un sistema operativo móvil y que se desarrollan en el lenguaje específico (nativo) de este sistema.

El uso de este tipo de aplicación obligaría al equipo a multiplicar el trabajo a la hora de desarrollar una app multiplataforma, puesto que cada sistema operativo requeriría una aplicación diferente, aumentando así el tiempo y el coste del trabajo. A pesar de ello, esto se plantea como trabajo a futuro.

2.3.4 Conclusión

En línea con lo expuesto en los apartados dedicados a las aplicaciones de escritorio y a las aplicaciones web, se determinó realizar una aplicación web, tanto para los clientes como para los administradores.

Si bien también podría haber sido adecuado realizar una aplicación de escritorio para el uso en el centro de copiado, se decidió realizar una misma aplicación web para ambos usos, para simplificar el desarrollo y el despliegue de la misma, además de las ventajas que presentan naturalmente las aplicaciones web. En el presente proyecto los aspectos más importantes son:

- Que no necesiten ser instaladas y que pueda ser accedida desde cualquier lugar y dispositivo, siendo esto gran parte lo que da valor al producto. Permitiendo que los nuevos usuarios no dependan de las condiciones de sus equipos para utilizarla (memoria o versión de sistema operativo).

- Una sola aplicación que puede ser utilizada desde casi cualquier plataforma y cualquier dispositivo reduce significativamente los costos y tiempos de desarrollo.

2.4 Cloud Computing

Cloud computing o computación en la nube es una tecnología que permite acceder remotamente, desde cualquier lugar del mundo y en cualquier momento, a software, almacenamiento de archivos y procesamiento de datos a través de Internet, sin la necesidad de conectarse a una computadora personal o servidor local.

La nube surgió como una forma de democratizar la información y mejorar la experiencia de quienes dependen de recursos tecnológicos a nivel personal o profesional.

El funcionamiento del cloud computing se basa en utilizar un servidor remoto para conectar los dispositivos de los usuarios a recursos centralizados. Un servidor remoto almacena todos los datos y programas necesarios, y puede estar ubicado en cualquier parte del mundo.

Las ventajas de optar por este servicio son simples de analizar:

- Reducción de costos de infraestructura: ya que no se necesita comprar hardware y software, así como sus gastos de instalación y mantenimiento.
- Economía de espacio: dado que los recursos se almacenan online.
- Centralización de la información: evita que los datos se mantengan en distintos programas, con distintas formas de acceso y autenticación.
- Aumento o disminución según la necesidad del cliente.
- Posibilidad de trabajo remoto.

3. Dominio

El dominio de la aplicación, tanto para los clientes como para los administradores, incluye principalmente:

- *Pedidos*: contienen la información del cliente, la fecha estimada de retiro del pedido, los archivos encargados, un email de contacto y el nombre del cliente.
- *Repositorio de archivos*: es una lista de archivos creada y actualizada por los administradores del sistema.
- *Archivos*: el origen de cada archivo se divide en:
 - Subido por el cliente.
 - Seleccionado del repositorio.
- *Administradores*: las personas que trabajan en la fotocopiadora obtienen usuarios registrados y gestionados mediante la plataforma externa "Auth0". Sólo el encargado o quien se designe en el centro de copiado podrá crear o eliminar usuarios de administración. Para acceder al panel de administración es necesario autenticarse con las credenciales de acceso.
- *Precios*: Determinan el valor de cada hoja de impresión según las características de la misma, como el color, el tamaño de la hoja y faz.

4. Herramientas utilizadas

4.1 Base de datos



En primer lugar, se decidió utilizar una base de datos relacional, ya que es el tipo de base de datos que ambos recursos del proyecto poseían conocimientos. A su vez, esta decisión fue revalidada en base al teorema de CAP, donde este mismo enuncia:

“Es imposible para un sistema de cómputo distribuido garantizar simultáneamente:

- *La consistencia (Consistency), es decir, cualquier lectura recibe como respuesta la escritura más reciente o un error.*
- *La disponibilidad (Availability), es decir, cualquier petición recibe una respuesta no errónea, pero sin la garantía de que contenga la escritura más reciente.*
- *La tolerancia al particionado (Partition Tolerance), es decir, el sistema sigue funcionando incluso si un número arbitrario de mensajes son descartados (o retrasados) entre nodos de la red.”*

Al ser este un sistema transaccional se prefiere consistencia y disponibilidad, resignando la escalabilidad que da la tolerancia al particionado.

Tomando en cuenta ello, las opciones fueron MySQL, SQL Server y PostgreSQL. La tercera opción fue la primera en no tenerse en cuenta, debido a que no representaba ventajas sobre las dos primeras y es de las que menos conocimientos se poseían, ya que sólo fue utilizada durante la materia “Gestión de Bases de Datos” y su respectivo trabajo práctico integrador. Por otro lado, los recursos poseen conocimientos y están habituados a trabajar con MS SQL Server, y también presentan conocimientos en proyectos pequeños realizados con MySQL. Ambas alternativas suplen las necesidades del proyecto, pero el principal determinante es el costo y la base de datos proporcionada por el servicio de hosting.

Por lo expuesto anteriormente y por la magnitud del proyecto se decidió utilizar la base de datos desarrollada por Oracle: MySQL.

Las principales características de MySQL son:

- Es open source, posee una versión gratuita y una versión comercial relativamente económica.
- Es multiplataforma, es decir que funciona en una amplia lista de sistemas operativos.
- Es uno de los gestores de bases de datos más utilizados en todo el mundo.
- Fácil instalación.
- Fácil administración.
- Alta performance debido a su simplicidad.
- Soporta múltiples motores de almacenamiento.

4.2 Backend



En cuanto al backend, se consideraron las opciones de utilizar los lenguajes:

- Node js: es un lenguaje muy utilizado actualmente, pero los recursos no poseen se encuentran familiarizados con el mismo.
- C#: ya que los recursos utilizan dicho lenguaje en sus ámbitos laborales y se encuentran familiarizados con el mismo.
- Java: porque fue el lenguaje utilizado por los recursos durante todos los proyectos universitarios y se poseen ciertos conocimientos en la tecnología.

Java es el lenguaje de programación más robusto, por ser tipado, manejar múltiples hilos y su fiabilidad probada gracias a más de 20 años de desarrollo. A su vez, es ampliamente usada en escala empresarial y es por estos motivos que se optó por utilizar este para el desarrollo del backend del proyecto.

Una vez definido el lenguaje, se procedió a buscar un framework adecuado para el proyecto, llegando a Spring Boot.

En rigor, Spring Boot es un módulo del framework Spring, que presenta características compatibles con el proyecto “sistema de gestión de impresiones para centros de copiado” y su arquitectura basada en microservicios, más específicamente para construir un API RESTful. Las principales ventajas del mismo, que lo hacen altamente productivo y fácil de utilizar, son:

- Facilita enormemente las tareas de configuración inicial para desarrollar dentro del ecosistema Spring, así como las tareas de despliegue.
- Es una plataforma de código abierto, gratuita y con una amplia comunidad de programadores detrás y se encuentra disponible documentación completa y bien estructurada.
- Permite desarrollar tanto webs como microservicios.
- Tiene todo lo necesario para abarcar las funcionalidades esperadas de cualquier aplicación profesional. Desde gestión de configuraciones, seguridad, capa de acceso a datos SQL y noSQL, inyección de dependencias, monitorización, testing, entre otras.

- El ecosistema de Spring es un ecosistema de módulos que está en constante evolución, los cuales se pueden añadir a cada aplicación según se necesite. Entre ellos se encuentran: Spring Boot (auto-configuración y despliegue), Spring Data (simplificación de acceso a los datos), Spring Security (autenticación y seguridad), y más.

El equipo de desarrollo cree que todas las características mencionadas hacen a Java y Spring Boot el lenguaje y framework adecuados al proyecto.

4.3 Frontend



El frontend es el servicio encargado de la interacción directa con el usuario, representa el lado visual del sistema y gran parte del éxito y aceptación del mismo. Un frontend bien desarrollado brinda al sistema una alta amigabilidad, productividad para los administradores, comprensibilidad y confiabilidad para nuevos usuarios.

Por este motivo el equipo de trabajo, en primera instancia, contempló dos opciones:

- Angular: Es un framework open source desarrollado por Google para facilitar la creación y programación de aplicaciones web SPA (Single Page Application). Basado en una estructura modular donde separa la lógica de la interfaz en cada componente y su lenguaje principal es Typescript. Posee todas las librerías necesarias para realizar una aplicación, pero pueden implementarse otras de terceros.
- React: Es una librería opensource de JavaScript creada por Facebook para desarrollar interfaces de usuario. Permite la creación de interfaces

de manera más limpia y ordenada, mediante componentes reutilizables. Su principal diferencia con Angular es que se depende de otras librerías para desarrollar una aplicación completa, pero tiene la ventaja de que ofrece un mejor rendimiento.

Luego de investigar y analizar las opciones, el equipo decidió que Next Js es la opción adecuada para este proyecto, un framework de JavaScript que permite crear aplicaciones web de React listas para salir a producción. El mismo posee diversos aspectos necesarios en el desarrollo de aplicaciones web disponibles, como ser el ruteo automático de componentes alojándose en una carpeta específica, renderizado en el servidor permitiendo la carga más rápido de las interfaces para el usuario, almacenamiento en caché automático.

Además este posee una curva de aprendizaje sencilla y dispone de mucha documentación oficial.

4.4 Procesador de pagos



Los procesadores de pago permiten a organizaciones de diversos tamaños procesar pagos con tarjeta de crédito o débito a través de Internet o interfaces de punto de venta (POS) tradicionales.

Si bien existen varias opciones, como TodoPago, UaláBis, PayU, Mobbex, entre otras, la pasarela de pagos elegida para el proyecto fue MercadoPago, por pedido del cliente.

MercadoPago es una de las plataformas más utilizadas en Argentina, soporta todas las opciones de pago: tarjetas de crédito y débito, transferencias bancarias, o pagos en efectivo mediante PagoFácil.

Al ser parte de una de las empresas más reconocidas del país, presenta una gran aceptación por parte de los clientes del negocio, una buena documentación e integración con los lenguajes de programación elegidos para este proyecto.

4.4 Otras herramientas utilizadas en el desarrollo

4.4.1 Integración continua



Git es un sistema de control de versión distribuida, lo que quiere decir que la base del código entero y su historial se encuentran disponibles en la computadora del desarrollador, lo cual permite un fácil acceso a las bifurcaciones, o branches en inglés, y fusiones , o merge en inglés. Es posible almacenar este repositorio en la computadora o utilizar una plataforma en línea como GitHub.

GitHub es una de las plataformas de trabajo colaborativo en línea más grandes del mundo, Git es un sistema de control de versión distribuida, lo que quiere decir que la base del código entero y su historial se encuentran disponibles en la computadora de todo desarrollador, lo cual permite un fácil acceso a las bifurcaciones y fusiones. Esencialmente, hace que sea más fácil para individuos y equipos usar Git como la versión de control y colaboración.

Los motivos para su uso incluyen que el equipo pueda trabajar al mismo tiempo, desde diferentes lugares sin preocuparse por sobrescribirse, así la seguridad de la plataforma y que los recursos ya se encontraban habituados a trabajar con la misma

4.4.2 Autenticación



Para la autenticación y gestión de usuarios de administración se optó por utilizar una plataforma tercerizada, dejando la seguridad en manos de una empresa especializada y confiable.

Para este proyecto se utilizará la solución de Auth0, una empresa que provee autorización, autenticación, gestión de usuario y roles, migración de usuarios, entre otras funcionalidades. Para ello utiliza OAuth2, un estándar abierto para implementar autorización.

Auth0 ofrece tanto planes de pago como un servicio gratuito, que es el que se utilizará para el desarrollo del sistema. También posee un excelente documentación, videotutoriales e integración con todos los lenguajes de programación.

Consultado en una entrevista sobre qué es Auth0 y a qué se dedica, uno de sus cofundadores respondió *“Para decirlo simple, hacemos la cajita de login. La que todos conocemos, la que usamos cuando ingresamos a un servicio en Internet, pago o no, en el que hay que registrarse. Todos odiamos los passwords: bueno, nosotros proveemos esa plataforma para hacerles el trabajo más fácil a los programadores y para que sea más seguro para los usuarios”*. Lo cual resume en gran parte los fundamentos de utilizar un proveedor externo.

4.4.3 Host



En el caso del hosting existen muchas opciones que se ajustan a las necesidades del proyecto, como Hostinger, Donweb, Ionos, entre otros. El equipo, en conjunto con el cliente, seleccionaron el proveedor nacional “Donweb” por su bajo costo, buen funcionamiento y facilidad de uso, así como por buenas experiencias anteriores del cliente, que desea continuar utilizando dicho proveedor.

4.4.4 Swagger



Swagger es un conjunto de herramientas de software open source para diseñar, construir, documentar, y utilizar los servicios API. Fue desarrollado por SmartBear Software e incluye documentación automatizada, generación de código, y generación de casos de prueba.

Swagger es la herramienta con mayor nivel de adopción en el mundo del desarrollo de software, por lo que permite crear documentación altamente comprensible. Es por ello que se puede decir que es el estándar de facto para las API reutilizables y mantenibles.

Adoptando Swagger en el proyecto se logró describir, consumir y visualizar la API, siendo de gran ayuda en las etapas de desarrollo para validar los endpoint generados sin depender del frontend o una aplicación como Postman.

Cuando se ejecuta el backend, esta genera una interfaz automáticamente donde se pueden visualizar y ejecutar los distintos endpoint. A su vez, brinda información detallada sobre cada uno de manera automática, alguna de ellas son el tipo de solicitud HTTP, ruta, parámetros de entrada, parámetros de salida, posibles excepciones, clases de dominio, entre otros. También se puede agregar información adicional con pocas líneas de código, como ser nombre, descripción detallada sobre la funcionalidad del endpoint y ejemplos de ejecución.

En conclusión es una herramienta que permite documentar el backend de manera sencilla y que además agiliza las actividades de pruebas o la posibilidad en un futuro de implementar endpoint para terceros.

4.5 Programas complementarios

4.5.1 IDE de Java



“Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No tiene en mente un lenguaje específico, sino que es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java usando el plug-in JDT que viene incluido en la distribución estándar del IDE.

Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.”

Sus principales características son:

- Perspectivas, editores y vistas: en Eclipse el concepto de trabajo está basado en las perspectivas o ventanas y editores, relacionadas entre sí, y que permiten trabajar en un determinado entorno de trabajo de forma óptima.
- Gestión de proyectos: el desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente, documentación, ficheros configuración, árbol de directorios.
- Depurador de código: se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente ayuda a mejorar el código.

- Extensa colección de plug-ins: están disponibles una gran cantidad de plug-ins, tanto publicados por Eclipse, como por terceros.
- Plug-in JDT: es el plug-in encargado del soporte del IDE al lenguaje Java, incluido en la versión estándar de Eclipse por defecto. Contiene las vistas vistas: Outline y Package Explorer, coloreado de código, autocompletado del texto con sugerencias dependientes del contexto, así como también configurar formateo de código, generar getters and setters, entre otras funcionalidades.

4.5.2 Editor de código fuente



Para el desarrollo de la UI, o interfaz gráfica de usuario, se utilizó el editor de código fuente “Visual Studio Code”.

Las principales características de este editor multiplataforma creado por Microsoft y que lo hacen uno de los más utilizados de su tipo son:

- Combina la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo
- Cuenta con herramientas de Debug hasta opciones para actualización en tiempo real del código en la vista del navegador y compilación en vivo de los lenguajes que lo requieran
- Tiene la opción de abrir una o más terminales locales en el mismo espacio de trabajo.

- Cuenta con una amplia biblioteca de extensiones, las cuales se traducen en un sin número de opciones para ser más eficiente a la hora de estar programando.
- Tiene la función de autocompletar para varios lenguajes
- Por último, pero no menos importante, el uso de esta herramienta es gratuito

4.5.3 Suite ofimática

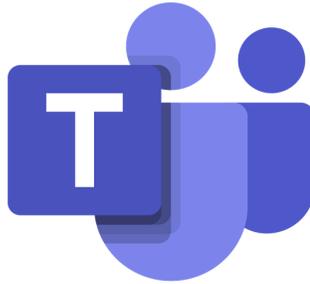


Las herramientas de trabajo que Google Workspace (conocida antes como GSuite) ofrece en su plan gratuito fueron las utilizadas como soporte para el desarrollo de informes, diagramas, etcétera.

Entre características, las que más se destacan en el ámbito de este proyecto son:

- Permite una colaboración en tiempo real entre grupos de trabajo
- Facilidad de uso
- Arquitectura 100% Cloud y multidispositivo
- Almacenar, sincronizar y compartir archivos
- Documentos disponibles y actualizados todo el tiempo
- Cero costo

4.5.4 Microsoft Teams



Por último, para las reuniones semanales, y recordando que el presente proyecto se da en el marco del proyecto final de carrera de los recursos para su título de grado en Ingeniería en Sistema de Información en la UTN Santa Fe, se utilizó la herramienta Microsoft Teams ya que se posee una licencia académica y es la herramienta que el equipo está acostumbrado a usar.

En la misma se creó un “equipo” que incluía a los dos participantes del proyecto y al director del proyecto, allí también se agendaron las reuniones para el seguimiento del proyecto, se compartieron recursos que persisten en el chat o sala del equipo, etcétera.

5. Desarrollo

En este apartado se detallan las estrategias, metodologías y resultados que se obtuvieron para la gestión y administración del proyecto.

A su vez, se detallan los requerimientos solicitados por el cliente y se mencionan cuáles fueron incluidos.

5.1 Requerimientos

Partiendo de la idea inicial del proyecto final presentado, se le propuso el sistema al cliente, con el cual luego se mantuvieron reuniones virtuales y presenciales

para continuar relevando los requerimientos según las preferencias y necesidades del mismo.

En dicho proceso, se realizó una serie de preguntas para comprender el funcionamiento del negocio y qué alcance podía tener la aplicación a desarrollar. Más adelante, en el apartado “extensibilidad” se detallan funcionalidades a implementar para una segunda versión del sistema.

También se consultó sobre los canales de comunicación digitales que utilizan para la toma de pedidos: si se lleva un registro de los mismos, qué medios de pago utilizan, si los pedidos recibidos vía medios digitales se imprimen antes de haber recibido el pago, entre otras.

Como resultado de este proceso se obtuvo la lista de requerimientos que se describe a continuación.

5.1.1 Lista de requerimientos

- A. Alta de usuario de administración: el dueño o encargado debe poder registrar nuevos usuarios administradores para los trabajadores del centro de copiado.
- B. Baja de usuario de administración: el dueño o encargado debe poder eliminar usuarios administradores de ex trabajadores del centro de copiado.
- C. Modificación de usuarios de administración: se deben poder modificar los datos, como email, nombre y contraseña de los usuarios administradores.
- D. Inicio y cierre de sesión de usuarios de administración: los usuarios administradores deben poder autenticarse utilizando sus credenciales para el uso del panel de administradores de la aplicación. También debe existir una función de cerrar la sesión.
- E. Alta de usuarios clientes: los clientes podrán registrarse como usuarios, disponiendo de un panel en el que se detallen todos los pedidos que realizaron.

- F. Modificación de usuarios clientes: los clientes podrán modificar sus datos de usuario, como nombre, correo electrónico y contraseña.
- G. Inicio y cierre de sesión de usuarios clientes: los clientes podrán autenticarse para así acceder al panel de modificación de usuario y al panel “Mis pedidos”.
- H. Creación de pedidos: desde la interfaz principal de la aplicación se debe poder ir a la sección “nuevo pedido”. En esta sección se podrá subir archivos y/o seleccionarlos del repositorio de archivos frecuentes y se podrá programar la fecha estimada de entrega del pedido, luego el sistema calculará el precio del pedido sumando el precio de cada documento solicitado y procederá a la pantalla de pago, donde se puede optar por pagar online mediante mercadopago o seleccionar la opción “pago en efectivo”. Luego de las tres etapas mencionadas, el pedido se guarda en la base de datos y se envía una notificación por mail a la fotocopidora.
- I. Programar pedido: Esta funcionalidad se deriva de la creación de pedidos, luego de subir los archivos, se debe seleccionar un horario estimado para retirar el pedido, este horario debe ser mayor al momento del pedido y debe estar dentro de los horarios de atención del local.
- J. Interfaz consultar estado pedido: desde la interfaz principal de la aplicación se debe poder ir a la sección “buscar un pedido”. En ella se deberá ingresar el número de pedido y se mostrará el estado del mismo.
- K. Cancelación de pedidos: el cliente podrá cancelar su pedido siempre y cuando el mismo no haya sido impreso aún. En el caso del local, un usuario administrador puede cancelar el pedido desde la interfaz de detalle del pedido en cualquier momento, excepto que ya haya sido entregado.
- L. Subir archivos: funcionalidad derivada de la creación de un pedido. En este apartado se podrán subir archivos .pdf o .doc desde el dispositivo en el que se haya ingresado a la aplicación. Luego de seleccionar el archivo, se seleccionan las características de impresión del mismo: simple o doble faz, color y tamaño de

hoja, y también se presenta un campo para detallar otras observaciones. El sistema calculará y mostrará el precio del archivo.

- M. Seleccionar archivos del repositorio de archivos frecuentes: esta funcionalidad también se deriva de la creación de pedidos, en ella se podrá filtrar la lista de archivos frecuentes proporcionada por el centro de copias, y seleccionar un archivo. Al seleccionarlo se muestra la cantidad de páginas del archivo seleccionado, se seleccionan las características de impresión y se calcula el precio.
- N. Gestión archivos frecuentes: panel al que sólo pueden acceder los usuarios administradores. En el mismo se presenta una lista de archivos frecuentes que puede filtrarse para buscar archivos. También se derivan las funcionalidades “alta de archivo frecuente” y “baja de archivo frecuente”.
- O. Alta de archivos frecuentes: En la misma interfaz de gestión de archivos se puede acceder a mediante un botón a seleccionar un archivo .pdf del dispositivo y añadirlo a la lista de archivos frecuentes.
- P. Baja de archivos frecuentes: Desde la lista de archivos frecuentes en el apartado “gestión de archivos frecuentes” se puede eliminar los archivos.
- Q. Calcular precio de pedidos: Esta funcionalidad calcula el precio de los archivos según sus preferencias de impresión y el precio base, mediante una fórmula proporcionada por el cliente.
- R. Procesador de pagos: Se deberá implementar el procesador de pagos “Mercado Pago” para el pago online de los pedidos. El mismo debe estar linkeado a una cuenta proporcionada por el cliente.
- S. Envío de correos electrónicos: En cada cambio de estado el pedido el sistema enviará automáticamente un correo electrónico:
 - Cuando el pedido es creado se envía un email al cliente y uno al centro de copias informando que el nuevo pedido fue creado exitosamente.

- Cuando el pedido es marcado como “impreso” por un usuario administrador, se le enviará un correo electrónico al cliente informando que su pedido ya está impreso y disponible para ser retirado
 - Cuando el pedido es marcado como “entregado” por un usuario administrador, se le enviará un correo electrónico al cliente informando que su pedido fue entregado
 - Si un usuario administrador decide cancelar un pedido, se le enviará un email al cliente informando que el pedido fue cancelado.
 - Si un cliente cancela un pedido, se le enviará un email al centro de copiado informando que el pedido fue cancelado.
- T. Lista de pedidos: Desde la interfaz principal de administradores se puede ingresar a la sección “pedidos” o bien a secciones según el estado del pedido: “pedidos pendientes”, “pedidos impresos” o “pedidos entregados”. En las mismas se listan los pedidos y algunos datos de los mismos. Desde allí se puede ingresar a ver el detalle de cada pedido, lo que da lugar a la siguiente funcionalidad: detalle de pedido.
- U. Detalle de pedido: En la sección de detalle del pedido, se pueden observar los datos asociados al cliente como nombre y email, así como los datos del pedido: estado, fecha estimada de entrega, medio de pago y la lista de archivos a imprimir junto con su especificación y botones para imprimir o abrirlo en otra pestaña.
- Esta funcionalidad es sólo para administradores.
- V. Impresión de documentos: desde la pestaña detalle de pedido se puede imprimir directamente en el navegador los archivos subidos en formato pdf.
- W. Actualización de precios base: dado el estado de inflación constante del país, se decidió crear una sección para actualización de precios base, los cuales están

determinados según la combinación de características de impresión: faz, tamaño de hoja y color.

5.1.2 Alcance del proyecto

Este proyecto comprende la primera versión totalmente funcional del sistema, la cual abarca las historias de usuario presentadas en la Tabla 6, junto con su estimación de esfuerzo, prioridad y riesgo.

Estas historias de usuario surgen en base a los requerimientos listados en el apartado anterior, y fueron consensuadas con el cliente.

ID	Rol	Historia de usuario	Esfuerzo	P	R
1	Admin	ABM de usuarios de administración	3	3	1
2	Admin	Inicio/cierre de sesión de usuarios de administración	3	3	1
3	Admin	ABM de usuarios	3	3	1
4	Usuario	ABM de pedidos	8	4	3
5	Usuario	ABM de archivos	13	4	3
6	Admin	AMB de archivos frecuentes	5	1	3
7	Admin	Integración del procesador de pagos	8	2	4
8	Admin	Gestor de alertas por correo electrónico	5	1	1
9	Admin	Gestionar pedidos	8	4	3
10	Admin	Gestor de impresoras	13	1	4
11	Usuario	Interfaz de principal	3	1	2
12	Admin	Interfaz inicial administrador	3	2	2
13	Admin	Gestor de precios	3	2	1

Tabla 6: Historias de usuario.

Total de esfuerzo = 75 puntos de esfuerzo.

Sobre la base de la experiencia en proyectos académicos y laborales previos, y en el esfuerzo estimado de cada historia, se realizó una definición de velocidad. Para esto, se utilizó una historia conocida por el equipo, como es el ABM de usuarios, que comprende tareas a las cuales los recursos están habituados. Si bien se verá luego que el alcance de la historia planificado originalmente fue modificado, el mismo fue utilizado para la estimación de horas de trabajo que requiere cada punto de historia. Así, para un ABM “clásico”, se estimó un esfuerzo de 3 puntos de historia se desarrollará aproximadamente en 8 horas de trabajo. Esto lleva al cálculo:

$$\frac{8 \text{ [horas de desarrollo]}}{3 \text{ [puntos de esfuerzo]}} \simeq 2:40 \frac{\text{[horas de desarrollo]}}{\text{[puntos de esfuerzo]}}$$

Dado que es una estimación, se redondeará el cálculo obtenido y se definirá la velocidad en 2:30 horas de trabajo por cada punto de historia, a fin de facilitar los cálculos siguientes.

Inicialmente se planificaron 7 Sprints, pero avanzado el desarrollo se revisó el alcance y la velocidad de desarrollo, y el equipo junto con el cliente decidieron:

- Eliminar el sprint 7 ya que abarcaba sólo un gestor de impresoras que no fue necesario gracias a la utilización de una librería de javascript para imprimir desde el navegador. Desde el inicio, esta historia se había planificado en el último sprint ya que se consideraba la posibilidad de tener que reducir el alcance, y la eliminación de esta última historia no afectaba demasiado el funcionamiento esencial de la aplicación. Se determinó agregar en la historia 9 una tarea para la implementación de la impresión de documentos.
- Debido a los problemas personales de los recursos que imposibilitaron cumplir las horas planificadas para el desarrollo del proyecto, y habiéndose extendido muchas semanas tanto en el primer como en el segundo sprint, se convocó a una reunión con el cliente, donde se decidió modificar las historias comprendidas en los sprints 3 y 4 relativas a registro y gestión de usuarios clientes, para unir ambos sprints en uno sólo.

De este modo, en el desarrollo real del sistema se realizaron 5 sprints, tal como se detalla en la siguiente sección.

5.2 Metodología

En el momento de la planificación del proyecto se optó por utilizar una metodología ad-hoc basada en SCRUM para el desarrollo de este sistema, con algunas variaciones para adaptarlo a los recursos que se poseen.

La metodología quedó definida con las siguientes características:

- División del desarrollo en Sprints.
- Stand-up meeting: dos reuniones semanales para evaluar avances, inconvenientes o planificar un cambio de tareas dentro del sprint
- Sprint Planning: reunión a principio del sprint para definir las tareas que se realizarán y quién estará a cargo de cada una.
- Sprint review: reunión al finalizar de cada sprint con el cliente.
- Product backlog: listado de tareas que engloba todo el proyecto.
- Sprint backlog: grupo de tareas del product backlog que se eligen en el sprint planning junto con el plan para poder desarrollarlas.

Teniendo en cuenta que el equipo sólo contaba con dos personas y una capacidad de 6:30 horas de trabajo disponibles semanalmente, resultando así un total aproximado de 27 hs mensuales, se procedió con la planificación como se expone a continuación.

Teniendo la historias de usuario refinadas y validadas con el cliente se realizó la estimación del esfuerzo de cada una de ellas contemplando el riesgo, dando como resultado un total de 75 puntos de esfuerzo.

Cada uno de estos puntos de esfuerzo fueron tomados en primera instancia con una equivalencia a 2:30 hs de trabajo, quedando así definido el primer sprint con 11 puntos de esfuerzos y un total de 27:30 hs.

Aunque en la práctica estos tiempos semanales no se cumplieron en todos los sprints se intentó ser lo más exacto posible sobre los tiempos dedicados a cada tarea para poder corregir la estimación en los siguientes sprints. Esto se logró colocando el las horas dedicadas a cada tarea, si bien no fue una métrica exacta fue de utilidad.

En cuanto a la relación con el cliente, en primera instancia, se generaron maquetas iniciales para el refinamiento de los requerimientos, luego durante el desarrollo se fueron dando reuniones breves para notificar avances, definir requerimientos que no fueron tenidos en cuenta o redefinir aquellos que estaban mal detallados.

Respecto a la división de tareas, ambos recursos oficiaron de desarrolladores full-stack, es decir, ambos realizaron tareas de backend y de frontend indistintamente. Así, en cada sprint se decidía qué tareas del sprint debía realizar cada uno.

La división natural que surgió y en la que las dos personas estuvieron de acuerdo fue que una desarrolle las funcionalidades para los clientes y la otra las funcionalidades para los administradores, es por ello que en los sprints suele balancearse las tareas que corresponden a cada lado de la aplicación.

A la vez, el trabajo fue muy colaborativo y en cada stand up meeting se exponían las complejidades que a cada recurso se le presentaban para poder resolverlas en conjunto, además de presentar los avances producidos por cada uno o plantear algún cambio de tareas.

A continuación se detallan los sprints realizados, junto con sus fechas de inicio y finalización, tareas y demás.

5.2.1 Sprint 1

El primer sprint comenzó el 22 de marzo y finalizó el 24 de mayo, con un total de 10 semanas.

Este sprint tenía como objetivo la configuración de los entornos de desarrollo, instalación de dependencias, sistemas de gestión de integración continua, la realización de la interfaz principal para los clientes y los administradores.

Como se puede ver en la Tabla 1, el sprint se excedió en tiempo por 6 semanas. Esto ocurrió porque se desconocía la tecnología elegida para realizar el frontend.

Como fue mencionado anteriormente, entender cómo configurar Next.js y aprender JavaScript requirió más tiempo del que se creía, así como configurar cada componente de la librería MaterialUi que se utilizó para los estilos.

Tareas	Semanas									
	22/03	29/03	05/04	12/04	19/04	26/04	03/05	10/05	17/05	24/05
Diseño inicial de la arquitectura de la aplicación	■	■								
Estructura inicial de la aplicación		■	■							
Configuración del servidor de integración continua y gestión de cambio			■							
H11: Interfaz principal				■	■	■	■	■	■	■
H11: Imágenes de guía								■	■	
H12: Interfaz administrador					■	■	■	■	■	■

Tabla 1: Sprint 1.

5.2.2 Sprint 2

El segundo sprint comenzó el 31 de mayo y terminó el 23 de agosto, dando así un total de 13 semanas (Tabla 2).

El objetivo de esta etapa fue desarrollar la funcionalidad principal del sistema, es decir, la gestión del pedido. La misma consiste en el alta, baja, modificación y consulta de un pedido que realiza un usuario de la imprenta. Cabe destacar que en esta implementación no se incluye la subida de los archivos y el pago del pedido, debido a

que esto significaba una complejidad mayor y se decidió desarrollarlo en la siguiente etapa.

Este sprint, igual que el inicial, se planteó de 4 semanas, pero se vio afectado por cuestiones personales de ambos integrantes del equipo, que los obligó a reducir su capacidad horaria semanal. También en este período se decidió realizar más capacitaciones en la tecnología seleccionada, ya que el bajo conocimiento que se poseía no permitía cumplir con la velocidad de desarrollo propuesta, ni las tareas asignadas a cada sprint.

Como consecuencia de lo mencionado anteriormente, en el sprint dos se decidió:

- Reducir la velocidad de desarrollo, para no cambiar el tamaño de las historias de usuario y poder ganar mayor velocidad en los siguientes sprints.
- Aumentar la ventana de tiempo del sprint hasta poder retomar el ritmo propuesto inicialmente, ya que se disponían muy pocas horas de trabajo efectivas.

Tareas	Semanas													
	31/05	07/06	14/06	21/06	28/06	05/07	12/07	19/07	26/07	02/08	09/08	16/08	23/08	
H4: Interfaz de realizar pedido	■	■	■	■	■	■	■	■	■	■	■			
H4: Interfaz de archivos disponibles												■	■	
H4: Backend ABMC de pedidos	■	■	■	■	■	■	■	■	■					
H4: Persistencia de pedidos en base de datos			■	■	■									
H4: Interfaz de estado de pedido													■	

Tabla 2: Sprint 2.

5.2.3 Sprint 3

El tercer sprint comenzó el 30 de agosto y terminó el 20 de septiembre, dando así un total de 4 semanas.

Esta etapa consistió en persistir la subida de archivos, perteneciente al proceso de creación de un pedido. A su vez, se agregaron las funcionalidades para visualización de los mismos en la interfaz del administrador según su estado y la posibilidad de impresión de cada uno de los archivos que lo componen.

También se corrigieron errores del sprint anterior y modificaciones sugeridas por el cliente.

Como muestra la Tabla 3, este sprint no se excedió de lo planificado en semanas. Esto no se debe a que las tareas fueron más sencillas que las anteriores, sino a que se aumentó la velocidad del sprint y se aumentaron las horas disponibles semanales de cada recurso, gracias a que se logró comprender la tecnología de frontend a un nivel que permitió al equipo cumplir con la planificación.

Tareas	Semanas			
	30/08	06/09	13/09	20/09
H5: Persistencia de archivos en la base de datos				
H5: Persistencia de archivos en el file system				
H5: Front subir un archivo en pedido				
H5: Backend amb archivos				
H9: Interfaz de pedidos pendientes				
H9: Interfaz de pedidos recibidos				
H9: Interfaz de impresión				
H9: Imprimir documento				
H9: Actualizar estado pedido				

Tabla 3: Sprint 3.

5.2.4 Sprint 4

El cuarto sprint comenzó el 27 de septiembre y finalizó el 11 de octubre, dando así un total de 3 semanas.

En esta etapa se agregó la autenticación de usuarios para la sección de administradores. A su vez, se implementó el pago por MercadoPago, dejando de esta manera finalizado el proceso de creación de un pedido iniciado en el segundo sprint. También se configuró el correo electrónico para notificar a los clientes sobre el estado de su pedido.

Como muestra la Tabla 4, se estimó un tiempo de cuatro semanas, pero sólo se utilizaron tres. Esto ocurrió por una sobreestimación, ya que se desconocía cómo se debería implementar el procesador de pagos y el gestor de correo electrónicos.

También se puede concluir que en el presente sprint el equipo ganó velocidad gracias a un mejor conocimiento del dominio y de las tecnologías y herramientas utilizadas.

Tareas	Semanas			
	27/09	04/10	11/10	18/10
H2: Autenticación de usuarios de administración				
H2: Cierre de sesión				
H7: Implementación de API Mercado Pago				
H7: Configuración y linkeo de cuenta del propietario				
H8: Configuración de servicio de correo electrónico				
H8: Configuración de notificaciones				
H8: Formato y contenido de correo electrónico				

Tabla 4: Sprint 4.

5.2.5 Sprint 5

El quinto y último sprint comenzó el 18 de octubre y finalizó el 8 de noviembre, dando así un total de 4 semanas.

En esta etapa se desarrolló la interfaz para la subida de archivos frecuentes, tanto el backend como el frontend de la interfaz del administrador.

Como muestra la Tabla 5, el sprint no se excedió de su tiempo planificado.

Tareas	Semanas			
	18/10	25/10	01/11	08/11
H6: Interfaz subir archivos frecuentes	■			
H6: Persistencia de archivos frecuentes en la base de datos		■		
H6: Modificar archivos frecuentes de la base de datos		■		
H6: Modificar ubicación de los archivos frecuentes en el file system		■		
H6: Eliminar archivos frecuentes de la base de datos			■	■

Tabla 5: Sprint 5.

Al finalizar este sprint se procedió a la redacción del presente informe, la cual requirió dos semanas más, iniciando la semana del 15 de noviembre y finalizando en la semana del 22 de noviembre.

5.3 Riesgos

El riesgo se define como la exposición a una situación donde hay una posibilidad de sufrir un daño o un evento desafortunado. Es la proximidad o amenaza a que ocurra un evento y sus efectos sean negativos y que alguien o algo, en este caso el proyecto, puedan verse afectados por él.

Sin embargo, los riesgos pueden reducirse o manejarse, por ello siempre es necesario un plan de gestión de riesgos que permita prevenirlos y/o mitigarlos.

En pos de realizar un plan de gestión de riesgos, primero se realizaron tareas de para identificar a los riesgos que se estaba expuesto, luego los mismos se analizaron y se clasificaron, llegando al siguiente resultado:

Riesgos de RRHH

#	Descripción del riesgo	Disparador
1	Imposibilidad de cumplir con la cantidad de horas de trabajo semanales de los recursos	Ausentismo a las reuniones, no cumplimiento de tareas planificadas
2	Un integrante abandona el proyecto	Despreocupación por avanzar en sus tareas
3	Desacuerdo en la toma de decisiones	Falta de comunicación
4	Conflicto interno del equipo	Opiniones adversas sobre temas de la actualidad o problemas en la toma de decisiones del proyecto

Tabla 7: Riesgos de recursos humanos.

Riesgos de desarrollo

	Descripción del riesgo	Disparador
5	Desconocimiento sobre las tecnologías	Excesivo tiempo al intentar implementar una funcionalidad
6	Pérdida de cambios por descuidos o desperfectos electrónicos.	Archivos faltantes en un check al servidor de integración o inestabilidad eléctrica los días de alta temperatura
7	Actualización de APIs e incompatibilidad con otros componentes	Dificultad al encontrar librerías oficiales para cumplir con los requerimientos

Tabla 8: Riesgos de desarrollo.

Riesgos de requerimientos

#	Descripción del riesgo	Disparador
8	Interpretación incorrecta de requerimientos	Están siendo gestionados por un sólo miembro del equipo
9	Inestabilidad de los requerimientos	El cliente no está seguro o no tiene una decisión final al momento de la toma de requerimientos
10	Especificación de requerimientos insuficiente o ambigua	Se realizan con poco tiempo disponible

Tabla 9: Riesgos de requerimientos.

Riesgos de performance

#	Descripción del riesgo	Disparador
11	Velocidad de respuesta del sistema no aceptable	La carga de datos es mayor a la estimada.
12	Espacio de almacenamiento limitado	Desinterés del cliente en contratar un servidor estimado para ahorrar costos

Tabla 10: Riesgos de performance.

Riesgos de diseño

#	Descripción del riesgo	Disparador
13	Baja amigabilidad del sistema	El flujo de pruebas es complejo
14	El sistema no cumple con los objetivos	El cliente solicita cambios de manera frecuente al finalizar cada sprint

Tabla 11: Riesgos de diseño.

Riesgos del producto

#	Descripción del riesgo	Disparador
15	Fiabilidad del producto	Los clientes siguen enviando con frecuencia los archivos por otros medios, como ser email o whatsapp
16	Seguridad en los pagos	Los clientes optan por el pago en efectivo de manera frecuente
17	Seguridad de los archivos	El servicio ftp contratado pierde o no renueva los certificados de seguridad con frecuencia o se capturan muchos errores al subir un archivo
18	Seguridad de los datos personales de los clientes	El sitio web no cuenta con los certificados de seguridad

Tabla 12: Riesgos del producto.

Riesgos del proceso

#	Descripción del riesgo	Disparador
19	Falla en la estimación de esfuerzo o velocidad para cada sprint	Tareas no completadas al finalizar el sprint
20	Excesiva detección de errores	Al menos un tercio de las pruebas no son completadas exitosamente

Tabla 13: Riesgos del proceso.

Luego de haber definido y analizado los riesgos, se definió la probabilidad de ocurrencia según la categorización que presenta la Tabla 14.

Categoría	Probabilidad
1	Ínfima
2	Baja
3	Media
4	Alta

Tabla 14: Categorización de probabilidad de ocurrencia.

Y el impacto potencial de los riesgos según la categorización presentada en la Tabla 15:

Categoría	Impacto
1	Despreciable
2	Considerable
3	Moderado
4	Crítico

Tabla 15: categorización de impacto.

Tanto las probabilidades de ocurrencia como el impacto potencial fueron determinados en base a las características del equipo, del proyecto y del cliente.

A partir de estos datos se calculó la exposición al riesgo (ER) mediante la fórmula:

$$ER = P * I$$

En la siguiente tabla se muestran los riesgos identificados junto a su categoría, probabilidad P, impacto I, ER, la estrategia que se adoptará frente a la aparición de los mismos y sus respectivas medidas de contingencia.

La estrategia definida para la mitigación de cada riesgo fue consensuada por los miembros del equipo, según su criterio y la importancia relativa que se le dió a cada

uno. En la medida de lo posible, se intenta reducir la probabilidad de ocurrencia, y cuando ésto no fuera posible, se decidió en la mayoría de los casos una aceptación activa.

En cuanto a los riesgos asociados al producto y los de performance, la estrategia fue decidida en conjunto con el cliente.

Num	Descripción	P	I	ER	Estrategia	Medidas de contingencia
1	Imposibilidad de cumplir con la cantidad de horas de trabajo semanales de los recursos	2	3	6	Atenuar	Diseñar un nuevo cronograma adaptado a la disponibilidad real de cada recurso
2	Un integrante abandona el proyecto	1	4	4	Aceptación activa	Revisión del alcance y cronograma del proyecto
3	Desacuerdo en la toma de decisiones	2	2	4	Aceptación activa	Establecer una metodología para la toma de decisiones. Consultar con los directores de proyecto sobre cuál es la mejor alternativa.
4	Conflicto interno del equipo	1	3	3	Aceptación activa	Resolver profesionalmente los conflictos.
5	Desconocimiento sobre las tecnologías	4	3	12	Aceptación activa	Realizar capacitación extra. Intercambiar experiencias y cooperar con los integrantes del equipo.
6	Perdida de cambios por descuidos o desperfectos electrónicos	1	2	2	Aceptación pasiva	
7	Actualización de APIs e incompatibilidad con otros componentes	1	2	2	Reducir	Usar versiones LTS
8	Interpretación incorrecta de requerimientos	2	3	6	Aceptación activa	Validar los requerimientos definidos
9	Inestabilidad de los requerimientos	2	3	6	Reducir	Reuniones en hitos de proyecto para validar requerimientos y entregas

						parciales
10	Especificación de requerimientos insuficiente o ambigua	3	2	6	Reducir	Validar los requerimientos definidos
11	Velocidad de respuesta del sistema no aceptable	2	2	4	Aceptación activa	Implementación de buenas prácticas y algoritmos eficientes
12	Espacio de almacenamiento limitado	2	1	2	Reducir	Evitar redundancia de archivos Mantenimiento periódico de la base de datos
13	Baja amigabilidad del sistema	2	2	4	Reducir	Validar cada entregable con el cliente Capacitación a usuarios
14	El sistema no cumple con los objetivos	2	3	6	Reducir	Validar cada entregable con el cliente
15	Fiabilidad del producto	2	3	6	Reducir	
16	Seguridad en los pagos	2	3	6	Reducir	Uso de APIs seguras Encriptación
17	Seguridad de los archivos	2	2	4	Reducir	Encriptación Implementar seguridad en la base de datos
18	Seguridad de los datos personales de los clientes	1	3	3	Reducir	Encriptación
19	Falla en la estimación de esfuerzo o velocidad para cada sprint	3	3	9	Aceptación activa	Estimación de esfuerzo consciente
20	Excesiva detección de errores	2	3	6	Aceptación activa	Pruebas Re-desarrollar partes con demasiados errores

Tabla 16: riesgos y medidas de contingencia

5.3.1 Riesgos ocurridos

De los veinte riesgos identificados al inicio del proyecto, ocurrieron cuatro de los riesgos con mayor probabilidad de ocurrencia. Los mismos fueron:

- Desconocimiento sobre las tecnologías: dado que para el desarrollo del frontend del sistema se decidió utilizar NextJs (React), una tecnología que ambos recursos desconocían, existía una alta probabilidad de que ocurra este riesgo. Al ocurrir, el mismo se aceptó, lo cual implicó consumir más tiempo del que estaba previsto en los primeros sprints, por lo que ambos participantes del desarrollo compartieron recursos para aprender y realizaron un curso en conjunto, tal como se había previsto en el plan de contingencia realizado.
- Imposibilidad de cumplir con la cantidad de horas de trabajo semanales de los recursos: debido a cuestiones personales de ambos recursos, los mismos se vieron imposibilitados de cumplir con las horas de trabajo previstas. Frente a esta situación, se optó por adaptar el cronograma a la disponibilidad horaria mientras duren los contratiempos.
- Falla en la estimación de esfuerzo o velocidad para cada sprint: la ocurrencia de este riesgo se relaciona con la ocurrencia de los dos riesgos mencionados previamente, ya que la velocidad de desarrollo fue menor y tampoco se pudo cumplir con los tiempos. Frente a esto, se ajustó la velocidad de los siguientes sprints, primero reduciendo la velocidad de desarrollo planteada y luego aumentando la misma, cuando el equipo se volvió más productivo por un mejor conocimiento de las tecnologías.
- Especificación de requerimientos insuficiente o ambigua: por último, se presentaron algunas situaciones en las que la especificación de requerimientos era ambigua, por lo que se planificaron reuniones con el cliente a fin de despejar dudas y validar los requerimientos.

6. Producto

El objetivo del sistema es permitir a los clientes de “Total Impresión Digital” realizar sus pedidos de impresión desde la web y abonarlos, si desean, desde su computadora o celular. No sólo se busca brindar la comodidad al cliente, sino, también ayudar a la gestión de la empresa centralizando los pedidos, cobros y organización de los archivos. Además, se pretende reducir los tiempos de espera que se generan al momento de realizar un pedido de impresión, centralizar los medios por los cuales llegan los archivos de los usuarios a las computadoras de la empresa y brindar diversos medios de pagos.

El sector empresarial que utiliza el sistema adquiere la ventaja de poder visualizar los pedidos que se encuentran pendientes en su orden de prioridad, imprimirlos y notificar al cliente que el mismo está listo para ser retirado. También, estos pueden visualizar pedidos en otros estados, permitiendo de esta manera tener una trazabilidad de las tareas realizadas.

Por otra parte, del lado del cliente, brinda la comodidad de no tener que desplazarse al centro de copiado para realizar un pedido, puede visualizar los horarios de atención al público y seleccionar una fecha de retiro que le parezca conveniente. A su vez, el mismo será notificado cuando esté listo.

También existe la funcionalidad de “archivos frecuentes”, esto permite a los usuarios administradores disponibilizar un archivo para todos los clientes, es decir, cualquiera que quiera realizar un pedido puede seleccionar éste e incluirlo. De esta forma, por ejemplo, los profesores pueden poner a disposición una misma versión de un apunte para todos sus alumnos, y a éstos encontrar el mismo de una manera mucho más accesible.

En las siguientes secciones se detalla cómo fue planteada la solución, desde la arquitectura de la misma hasta el listado de funcionalidades.

6.1 Arquitectura

La idea general de la aplicación fue desde un principio mantener una arquitectura sencilla pero escalable. Como se puede ver en la Figura 1, sólo se tienen dos servicios, uno para el frontend y otro para el backend, donde ambos tipos de usuarios interactúan con el primero, pero con distintas credenciales.

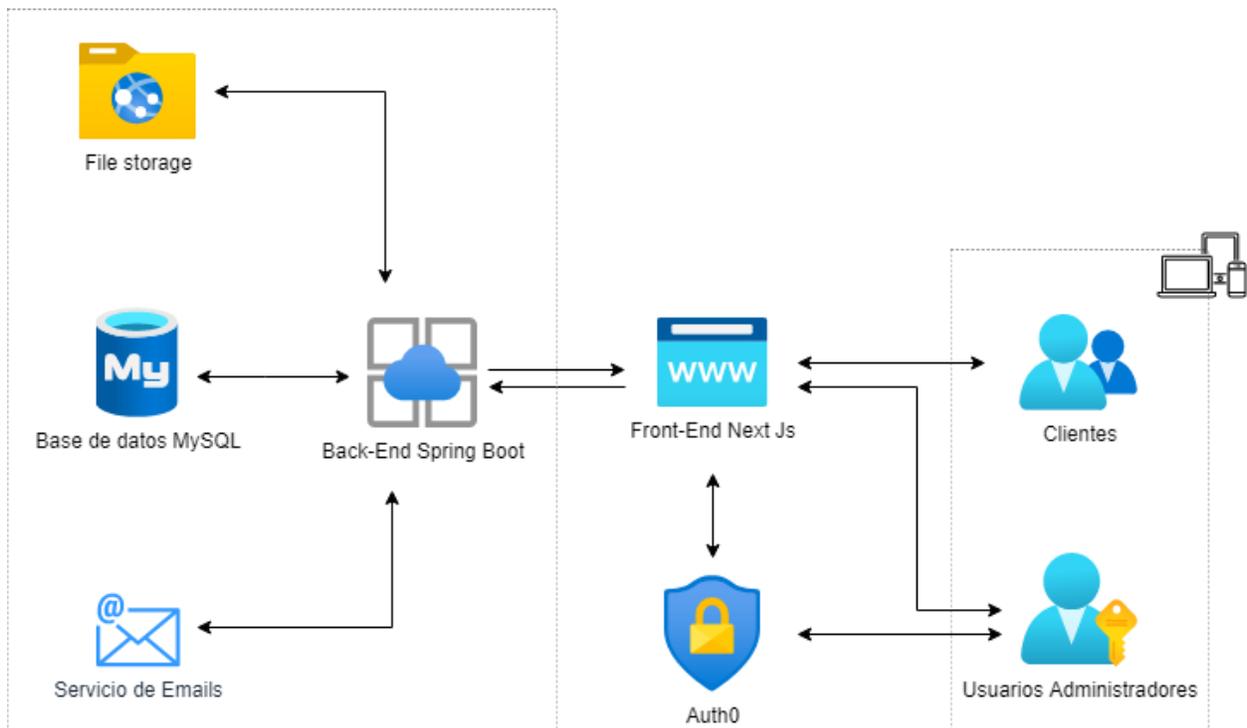


Figura 1 - Arquitectura de "Total Impresión Digital".

En la Figura 1 se observa un servidor central de backend, encargado de interactuar con la base de datos, el servicio de correo electrónico y el file storage.

El file storage es el encargado de almacenar los archivos que se suben en el sistema. Actualmente es un directorio dentro del mismo servidor, pero su interfaz está diseñada para implementar un servidor FTP. Esta decisión en relación a que sea una carpeta del mismo servidor, fue tomada por motivos económicos, ya que el cliente

decidió disminuir los costos hasta que los usuarios empiecen a utilizar el sistema con mayor frecuencia.

La arquitectura posee otro servicio encargado del frontend, en el cual interactúan todos los usuarios, ya sean administradores o clientes. La parte que visualiza el cliente es 100% adaptable a la pantalla del dispositivo, logrando una aplicación compatible para smartphones, computadoras o cualquier dispositivo con conexión a internet y un navegador.

El frontend interactúa con el servicio de Auth0 para obtener las credenciales de autenticación de los usuarios administradores. Ésta le brinda los permisos necesarios para navegar dentro de las opciones de gestión empresarial.

6.2 Modelo de datos

Comenzado el proyecto, se desconocían tanto las entidades a utilizar con exactitud, como así también las relaciones que podrían existir. Por esto, en un principio, se crearon las entidades básicas que luego se fueron modificando y agregando debido al desarrollo incremental.

A continuación, en la Figura 2, se expone el modelo inicial y luego, en la Figura 3, el diagrama final resultante de las clases de dominio.

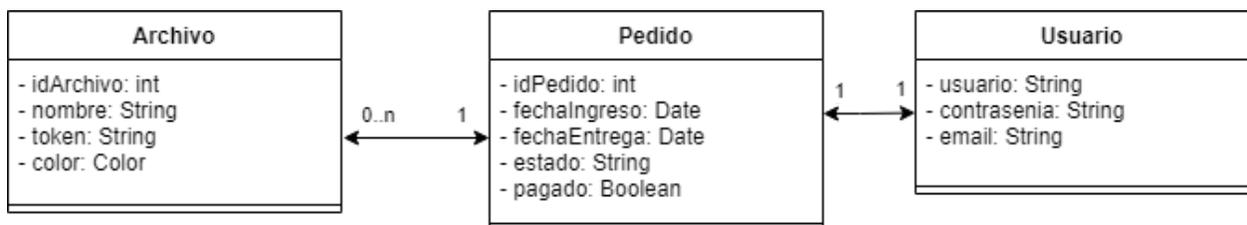


Figura 2: Modelo de clases inicial.

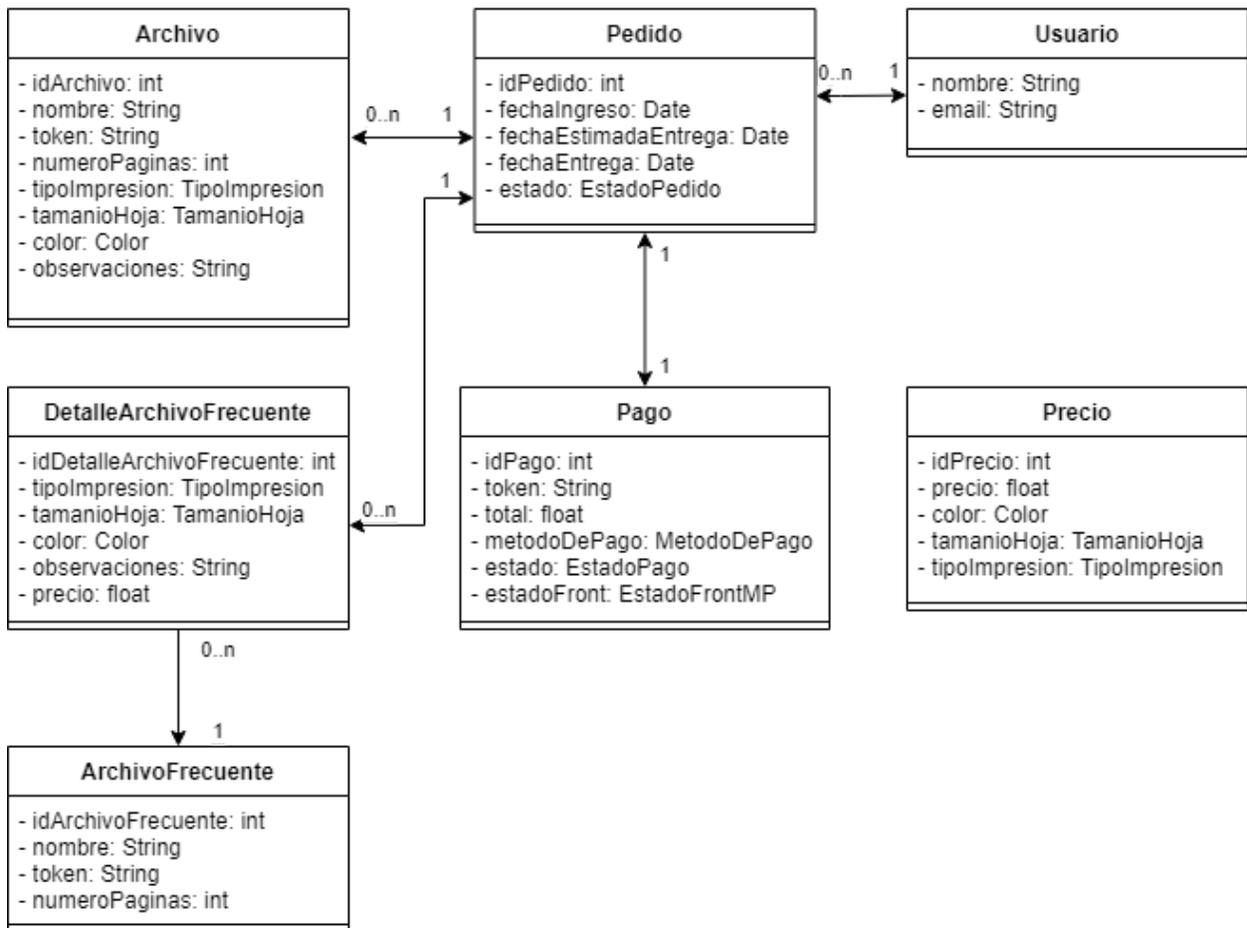
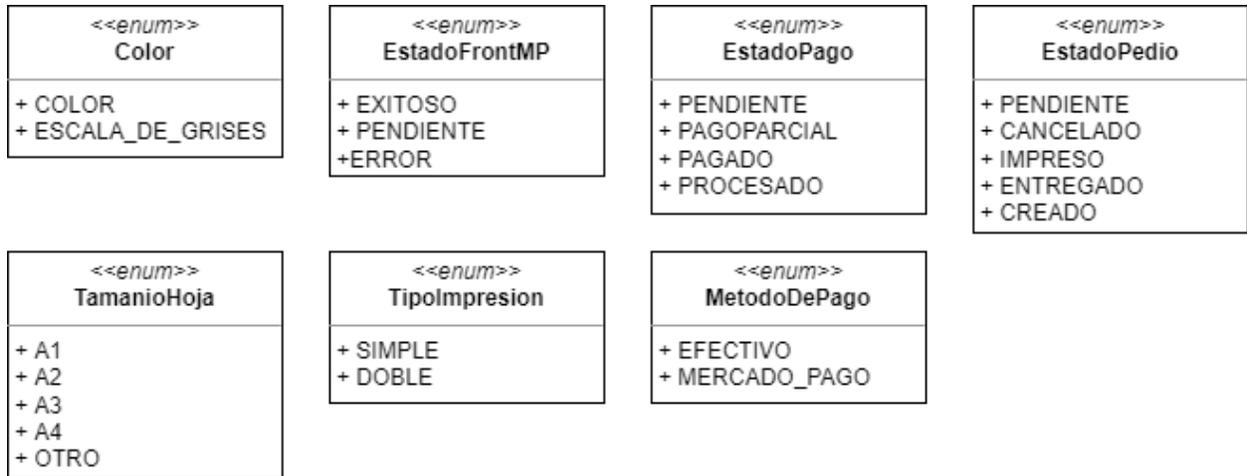


Figura 3: Diagrama de clases final.

Como se puede observar, en el diagrama final (Figura 3), la entidad protagonista es Pedido, siendo ésta aquella que crea el cliente en el proceso principal de la aplicación.

También se puede observar una entidad Usuario que actualmente sólo almacena el email y nombre de los clientes en el proceso de crear un pedido. Está pensada para que en futuras implementaciones sea la tabla donde se autenticuen los clientes para poder gestionarlos.

A su vez, la clase Pedido posee dos colecciones de archivos. Una de ellas pertenece a la lista de archivos que sube el cliente y la otra a archivos que se seleccionan del repositorio de “archivos frecuentes”. Cada uno de éstos posee un conjunto de atributos que representan la configuración (formato de impresión, tamaño de hoja u observaciones), otros que son los que lo caracterizan (cantidad de páginas, nombre, token) y el precio que se calcula dependiendo de estos atributos.

Las clases Archivo y Archivo frecuente poseen la mayoría de sus atributos en común, lo que indica que deberían ser la misma clase. Esto fue detectado por el equipo, pero la implementación ya estaba muy avanzada, por ende, realizar esta modificación implicaría mucho tiempo de desarrollo que no estaba disponible dentro del sprint, pero se prevé modificarlas en próximas iteraciones.

La clase Precio representa el precio que tendrá una hoja de un archivo con una determinada configuración, de ella se obtiene el precio que se fija en las clases Archivo y DetalleArchivoFrecuente.

6.3 Estructura backend

El backend se encuentra desarrollado con un diseño basado en interfaces, lo que permite abstraer la lógica de negocio de las distintas herramientas implementadas, en cualquier instancia del proyecto, sólo con cambiar la implementación.

Esta acción mencionada anteriormente suele realizarse en las bases de datos, por ejemplo, actualmente el sistema trabaja con MySQL pero en un futuro se podría

migrar a MongoDB y simplemente cambiando la implementación, se tendría todo el sistema funcionando.

En la Figura 4 se puede observar el funcionamiento básico de la estructura de backend, donde las peticiones entran por las clases Controller y son derivadas a las implementaciones del servicio mediante la interfaz. Éstas son las que modifican y crean las entidades de dominio y, para guardarlas o recuperarlas, lo hacen mediante la interfaz del repositorio.

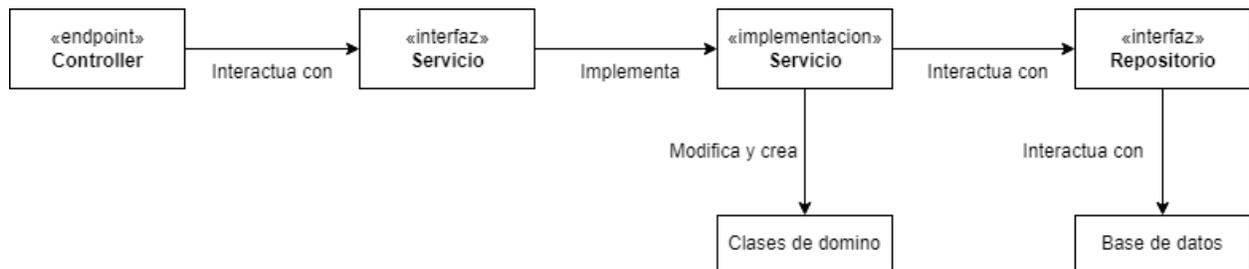


Figura 4: Funcionamiento backend.

En la Figura 5 se muestra cómo se organizaron las clases dentro de la solución del backend, donde se pueden observar tres grandes grupos.

En el llamado *Controller* se encuentran los componentes que son aquellos que responden a la interacción que hace el usuario en la interfaz y realiza los servicios correspondientes. Se creó uno por cada clase de modelo de negocios, el cual únicamente realiza acciones y responde con objetos de esa misma clase.

Dentro de la carpeta *Infraestructura* se encuentran el conjunto de clases necesarias para ejecutar los distintos procesos del sistema.

Dentro de la carpeta *Entidades* se encuentran las clases de modelo de negocio y los enumerables en su carpeta correspondiente, así como las clases DTOs. Estas últimas mencionadas son objetos de transferencia de datos (Data Transfer Object) y son utilizadas en la comunicación entre el frontend y el backend para evitar enviar datos innecesarios.

Por último, dentro de este directorio nombrado *Infraestructura*, cabe destacar las carpetas *Repositorios* y *Servicios*. Dentro de ellas se encuentran las interfaces de los

mismos, que son luego implementadas en las carpetas que están por fuera de infraestructura.

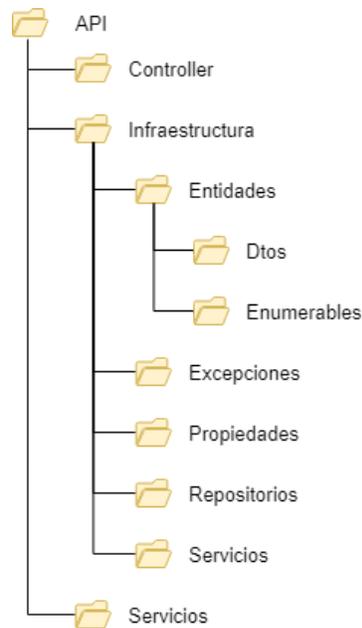


Figura 5: Organización de archivos de backend.

6.4 Funcionalidades

En el presente apartado se explicarán las funcionalidades de la aplicación junto con sus respectivas capturas de pantalla.

Para mejor claridad, las funcionalidades que pueden utilizar los clientes y los administradores se listan por separado, comenzando por aquellas a las que pueden acceder los clientes.

Todas las interfaces visibles a los clientes son 100% responsivas, es decir que se adaptan al tamaño de la pantalla del dispositivo desde el que se ingresa ya sea un smartphone, tablet o computadora de escritorio, por lo que cada interfaz de dicha sección se mostrará en sus versiones mobile y desktop.

En cuanto a las funcionalidades para los administradores, sólo está previsto que sean utilizadas desde computadoras de escritorio.

Funcionalidades para los clientes

Al ingresar a la aplicación, se encuentra la interfaz principal. En la misma se explican brevemente los pasos a seguir para realizar un pedido, ellos son:

- Subir o seleccionar archivos en formato .pdf
- Seleccionar las preferencias de impresión
- Pagar desde la web o en el local y retirarlo

También se presentan los botones “Realizar pedido” y “Buscar un pedido”, funcionalidades que se detallan más abajo.

Por último, en esta interfaz se muestran a modo informativo los precios de las principales formas de impresión.

Una solución moderna para tus pedidos de impresión

La forma más fácil de hacer tus impresiones

[REALIZAR PEDIDO](#)[BUSCAR UN PEDIDO](#)

Subí tu archivo

Podes seleccionar un archivo en formato .PDF de tu almacenamiento local o de tu catálogo de archivos frecuentes



Elegí las opciones

Podes elegir el tipo de impresión, color y papel

Retíralo

Retiras tu impresión pagando desde la web o en el local



Precios

Copia simple A4

\$5,00

Copia doble faz A4

\$8,00

Copia color A4

\$17,00

[VER MAS PRECIOS](#)

San Martin 1905 - Santo Tomé

+54 342 5066420

Horario de atención de 8:00 a 13:00 de Lunes a Viernes

Copyright © Total Impresión Digital 2021.

Figura 6: Vista desde un dispositivo de escritorio.



Figura 7: Vista desde un dispositivo móvil.

6.4.1 Realizar pedido

Al presionar el botón “Realizar pedido” de la interfaz principal, se muestra la interfaz para crear un pedido. En ella, se puede observar una lista de los archivos subidos o seleccionados, junto con la especificación de impresión de cada uno.

También se presentan los botones “Subir archivo” y “Seleccionar archivo del repositorio”.

Nuevo Pedido

1 Seleccionar Archivos — 2 Programar pedido — 3 Pagar

 AGREGAR ARCHIVO

SELECCIONAR DEL REPOSITORIO

Suba un archivo para continuar

ATRAS

SIGUIENTE



San Martin 1905 - Santo Tomé

+54 342 5066420

Horario de atención de 8:00 a 13:00 de Lunes a Viernes

Copyright © Total Impresión Digital 2021.

Figura 8: Vista nuevo pedido - desktop.

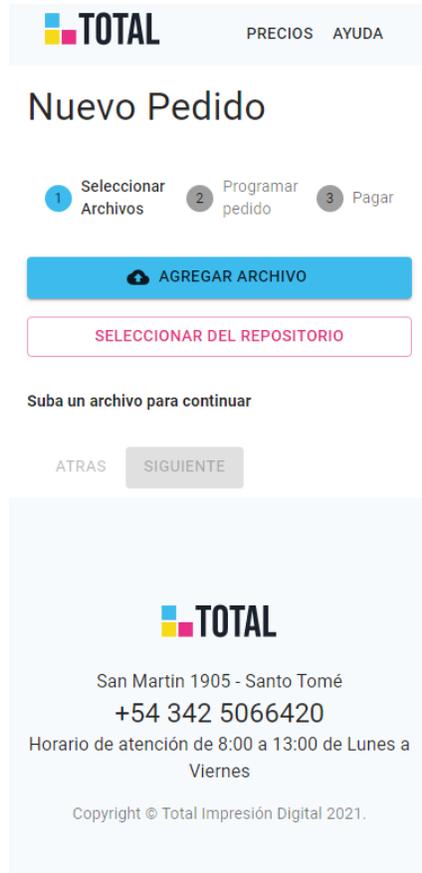


Figura 9: Vista nuevo pedido - mobile.

6.4.1.1 Subir archivo

Al presionar este botón se abrirá un cuadro de diálogo del sistema para seleccionar un archivo del dispositivo. Originalmente estaba previsto sólo permitir archivos en formato .pdf, pero por pedido del cliente se agregó la capacidad de recibir también archivos .docx, ya que algunos clientes desconocen cómo cambiar el formato.

Luego de seleccionar un archivo del dispositivo, se abrirá un cuadro de diálogo para especificar las preferencias de impresión. El mismo se explica más abajo y se muestra en la Figura 13 para la versión de escritorio y en la Figura 14 para la versión móvil.

Repositorio

Nombre del archivo 

Fisica 2do B.pdf
Profesora Maria Florencia

Coeficientes de fricción.pdf
Profesor Juan Carlos - Fisica 3

Tabla de derivadas.pdf
Parcial 2 5C

Telegrama de renuncia.pdf
Ley 1231

Parcial 2017 - Gestión de datos.pdf
Con respuestas

Parciales de ejemplo Análisis 2.pdf
Resueltos

 **1** 

Seleccione un archivo para agregarlo a su pedido.

CANCELAR

Figura 11: Seleccionar archivo frecuente - desktop.

Repositorio

🔍

Fisica 2do B.pdf
Profesora Maria Florencia

Coeficientes de fricción.pdf
Profesor Juan Carlos - Fisica 3

Tabla de derivadas.pdf
Parcial 2 5C

Telegrama de renuncia.pdf
Ley 1231

Parcial 2017 - Gestión de datos.pdf
Con respuestas

Parciales de ejemplo Análisis 2.pdf
Resueltos

<12>

Seleccione un archivo para agregarlo a su pedido.

CANCELAR

Figura 12: Seleccionar archivo frecuente - mobile.

Para finalizar con la inclusión de cada archivo en el pedido, se deben especificar las preferencias de impresión del mismo. Éstas incluyen:

- Color:
 - Escala de grises
 - Color
- Tipo de impresión:
 - Simple faz

- Doble faz
- Tamaño de hoja:
 - A1
 - A2
 - A3
 - A4
 - Otro

Como se puede observar en la Figura 13, también se provee un espacio de texto para especificar otras observaciones, en caso de ser necesario.

El cuadro mostrará también la cantidad de páginas y el precio calculado en base a las preferencias de impresión seleccionadas.

The screenshot shows a web interface for 'TOTAL' with a 'Nuevo Pedido' form. The form is titled 'Nuevo archivo' and displays the following information:

- Cantidad de páginas: 16
- Precio: \$115
- Formato: SIMPLE (dropdown menu)
- Color: ESCALA DE GRISES (dropdown menu)
- Tamaño de hoja: A3 (dropdown menu)
- Observaciones: (text area)

At the bottom of the form, there are two buttons: 'AGREGAR' (highlighted in blue) and 'CANCELAR'. The background of the page is dark gray with the 'TOTAL' logo in the top left and 'PRECIOS AYUDA' in the top right. A copyright notice 'Copyright © Total Impresión Digital 2021.' is visible at the bottom of the page.

Figura 13: seleccionar preferencias de impresión - desktop

Nuevo archivo

Cantidad de páginas: 16

Precio: **\$115**

Formato

SIMPLE

Color

ESCALA DE GRISES

Tamaño de hoja

A3

Observaciones

AGREGAR

CANCELAR

Figura 14 - Seleccionar preferencias de impresión - mobile.

Al finalizar, el archivo se incluirá en la lista y se habilitará el botón “Siguiente”, el cual nos lleva al siguiente paso: “Programar pedido”.

Nuevo Pedido

Numero de orden #247

1 Seleccionar Archivos — 2 Programar pedido — 3 Pagar

AGREGAR ARCHIVO

SELECCIONAR DEL REPOSITORIO

Archivo	Formato	Tamaño	Precio
Proyecto final - Raviolo Vola.pdf	SIMPLE	A4	\$420
Diagramas de clases.pdf	SIMPLE	A3	\$115

Total: \$1575

[ATRAS](#)

[SIGUIENTE](#)



San Martin 1905 - Santo Tomé

+54 342 5066420

Horario de atención de 8:00 a 13:00 de Lunes a Viernes

Copyright © Total Impresión Digital 2021.

Figura 15: Lista de archivos en el pedido - desktop.

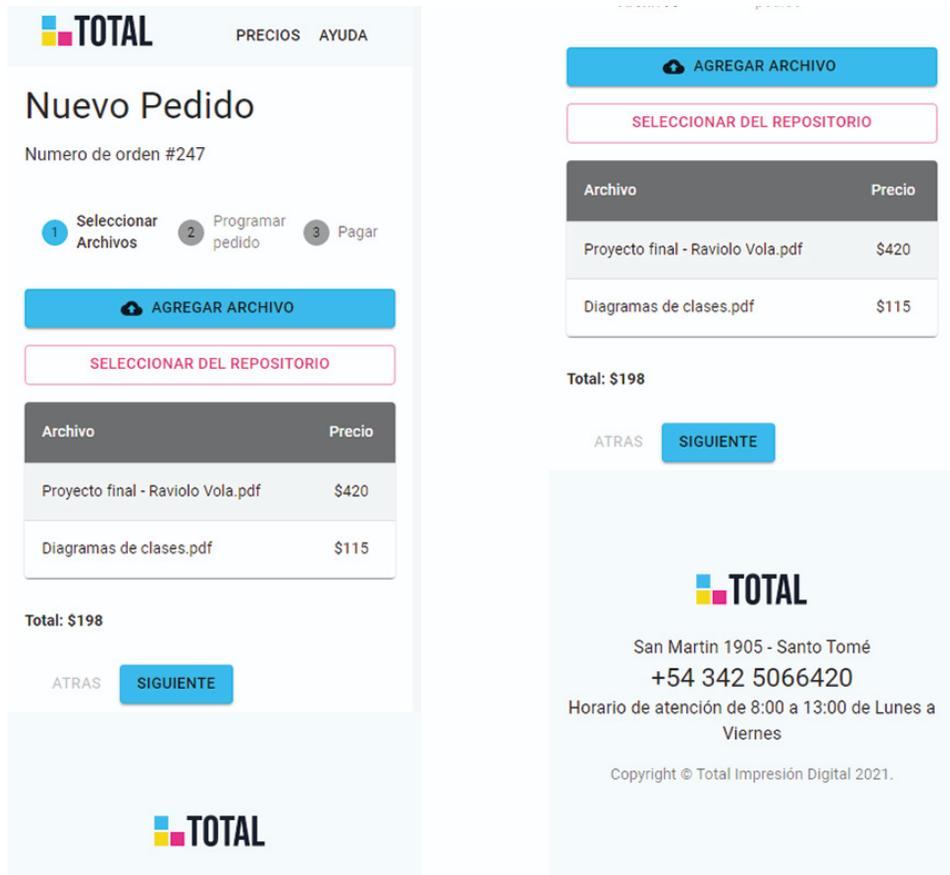


Figura 16: Lista de archivos en el pedido - mobile.

6.4.2 Programar pedido

El siguiente paso para realizar un pedido es programar la fecha y horario de retiro del mismo y agregar los datos del cliente: nombre y correo electrónico, como se puede observar en las Figuras 17 y 18.

La fecha y los horarios elegibles para el retiro de pedido, además de que deben ser posteriores al momento en que se realiza el pedido, se validan con los días y horarios de apertura y cierre de la atención al público en el local.

En la base de datos, si ya se tiene registrado un usuario con el correo electrónico provisto por el cliente, se relaciona el pedido a dicho usuario. Si no, se registrará un nuevo usuario.

Nuevo Pedido

Numero de orden #252

Seleccionar Archivos ———— **2** Programar pedido ———— **3** Pagar

Horario de retiro 

Lunes a Viernes de 8:00hs a 13:00hs

Correo electrónico *

Nombre y Apellido *

[ATRAS](#)

[SIGUIENTE](#)



San Martin 1905 - Santo Tomé

+54 342 5066420

Horario de atención de 8:00 a 13:00 de Lunes a Viernes

Copyright © Total Impresión Digital 2021.

Figura 17: Programar pedido - desktop.

Nuevo Pedido

Numero de orden #252

- Seleccionar Archivos** **2 Programar pedido** **3 Pagar**

Horario de retiro 

Lunes a Viernes de 8:00hs a 13:00hs

Correo electrónico*

Nombre y Apellido*

[ATRAS](#)

[SIGUIENTE](#)



San Martin 1905 - Santo Tomé

+54 342 5066420

Horario de atención de 8:00 a 13:00 de Lunes a Viernes

Copyright © Total Impresión Digital 2021.

Figura 18: Programar pedido - mobile.

6.4.3 Pagar pedido

Aquí, como se muestra en las Figuras 19 y 20, se puede optar por pagar online vía Mercado Pago o informar que se pagará en efectivo al momento de retirar el pedido.

TOTAL PRECIOS AYUDA

Nuevo Pedido

Numero de orden #252

✓ Seleccionar Archivos ——— ✓ Programar pedido ——— 3 Pagar

PAGAR CON MERCADO PAGO

Pagar en efectivo

ATRAS TERMINAR

TOTAL

San Martin 1905 - Santo Tomé
+54 342 5066420
Horario de atención de 8:00 a 13:00 de Lunes a Viernes

Copyright © Total Impresión Digital 2021.

Figura 19: Pagar pedido - desktop.

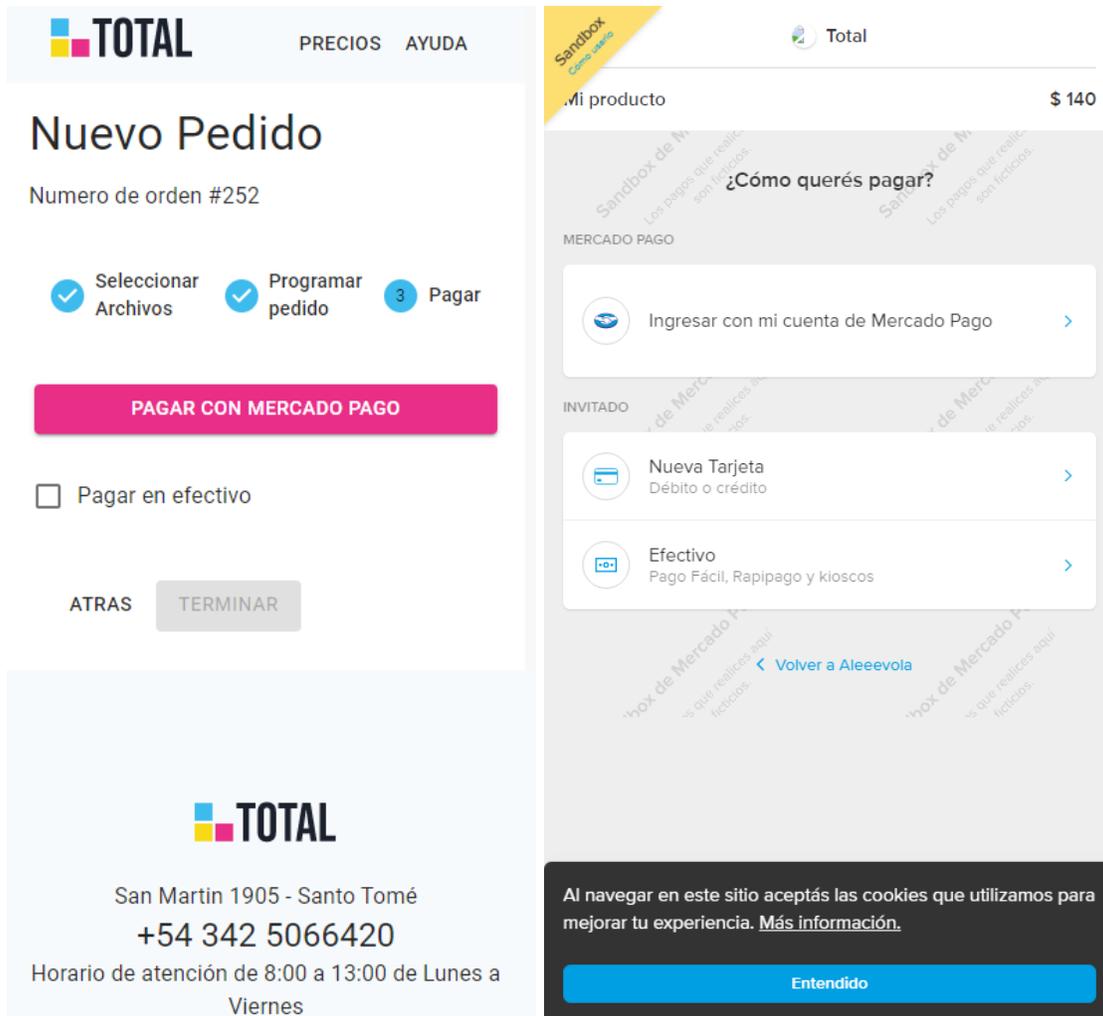


Figura 20: Pagar pedido - mobile.

6.4.4 Buscar pedido

Al presionar el botón buscar pedido presente en la interfaz principal, se mostrará una ventana emergente con un cuadro de texto para ingresar el número de pedido y un botón para realizar la búsqueda, tal como se muestra en la Figura 21.

Al no tener implementado el registro de usuarios clientes, en esta versión del sistema sólo se mostrará el número de pedido y el estado en que se encuentra, sin más datos del cliente por cuestiones de privacidad.

Una solución moderna para tus pedidos de impresión

La forma más fácil de hacer tus impresiones

REALIZAR PEDIDO

BUSCAR UN PEDIDO

Estado de pedido

Numero de pedido



CERRAR

Subí tu archivo

Podes seleccionar un archivo en formato .PDF de tu almacenamiento local o de tu catálogo de archivos frecuentes



Figura 21: Buscar pedido - desktop.

Funcionalidades para los administradores

6.4.4 Inicio y cierre de sesión

Al ingresar a la sección de administradores, si no se está autenticado, el sistema lo redireccionará al inicio de sesión de Auth0. Allí se ingresarán las credenciales de inicio de sesión. Auth0 validará las credenciales y redireccionará al usuario automáticamente a la interfaz principal de administración del sistema.

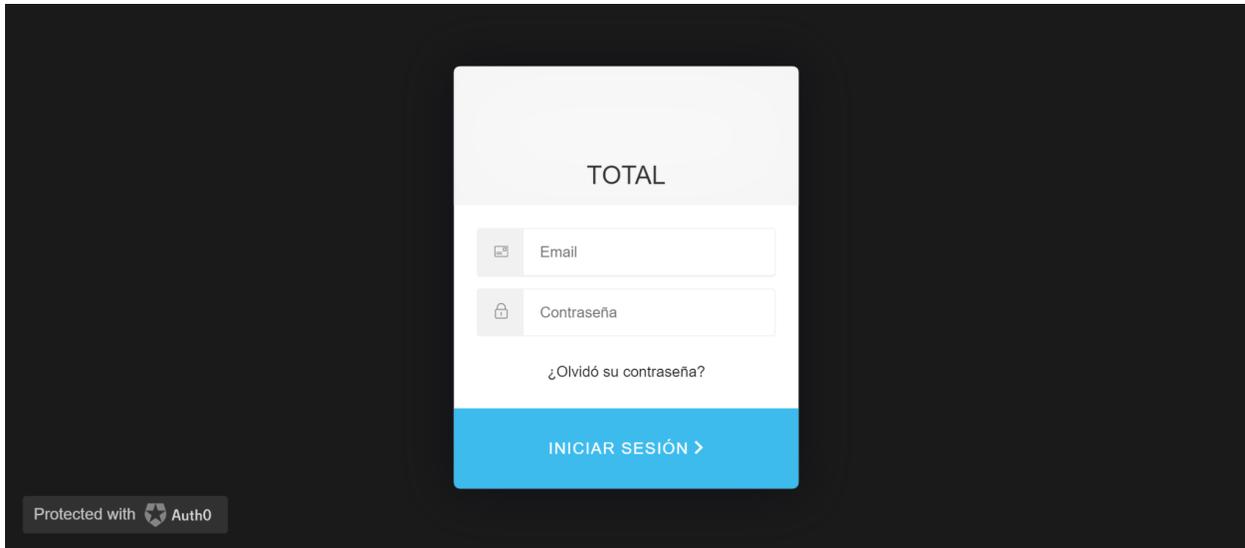


Figura 22: Inicio de sesión administradores.

En la interfaz principal se muestra un barra de navegación en la parte superior de la pantalla, donde se detalla el nombre del usuario autenticado, un dashboard o resumen con la cantidad de pedidos pendientes, la cantidad de pedidos impresos y entregados. Más abajo se encuentra el panel en el que se muestran en forma de tabla los precios actuales de cada hoja de impresión según sus características, junto con un botón para la actualización de los mismos. También se presenta una barra lateral, con los menús “Home”, “Pedidos” y “Gestión archivos”.



- HOME
- PEDIDOS
- GESTIÓN ARCHIVOS

PEDIDOS

PENDIENTES Hoy ▾ 3 pedidos, 15 páginas VER PEDIDOS PENDIENTES	IMPRESOS Hoy ▾ 2 pedidos, 36 páginas VER PEDIDOS IMPRESOS	ENTREGADOS Hoy ▾ 3 pedidos, 29 páginas VER PEDIDOS ENTREGADOS
---	---	---

PRECIOS

Precio	Color	Tamaño	Faz	
\$18	ESCALA DE GRISES	A4	SIMPLE	ACTUALIZAR
\$14	ESCALA DE GRISES	A4	DOBLE	ACTUALIZAR
\$22	ESCALA DE GRISES	A3	SIMPLE	ACTUALIZAR
\$25	COLOR	A4	SIMPLE	ACTUALIZAR
\$21	COLOR	A4	DOBLE	ACTUALIZAR
\$32	COLOR	A3	SIMPLE	ACTUALIZAR

Figura 23: Pantalla principal administradores.

6.4.5 Actualizar precios

Cuando se selecciona un precio para actualizar, se abrirá un cuadro de diálogo en el que se detallan las características de impresión del precio seleccionado junto con un cuadro de texto que permite especificar el nuevo valor. Al guardar, el nuevo valor será actualizado en la base de datos siempre y cuando se haya ingresado un valor entero o decimal mayor a cero.

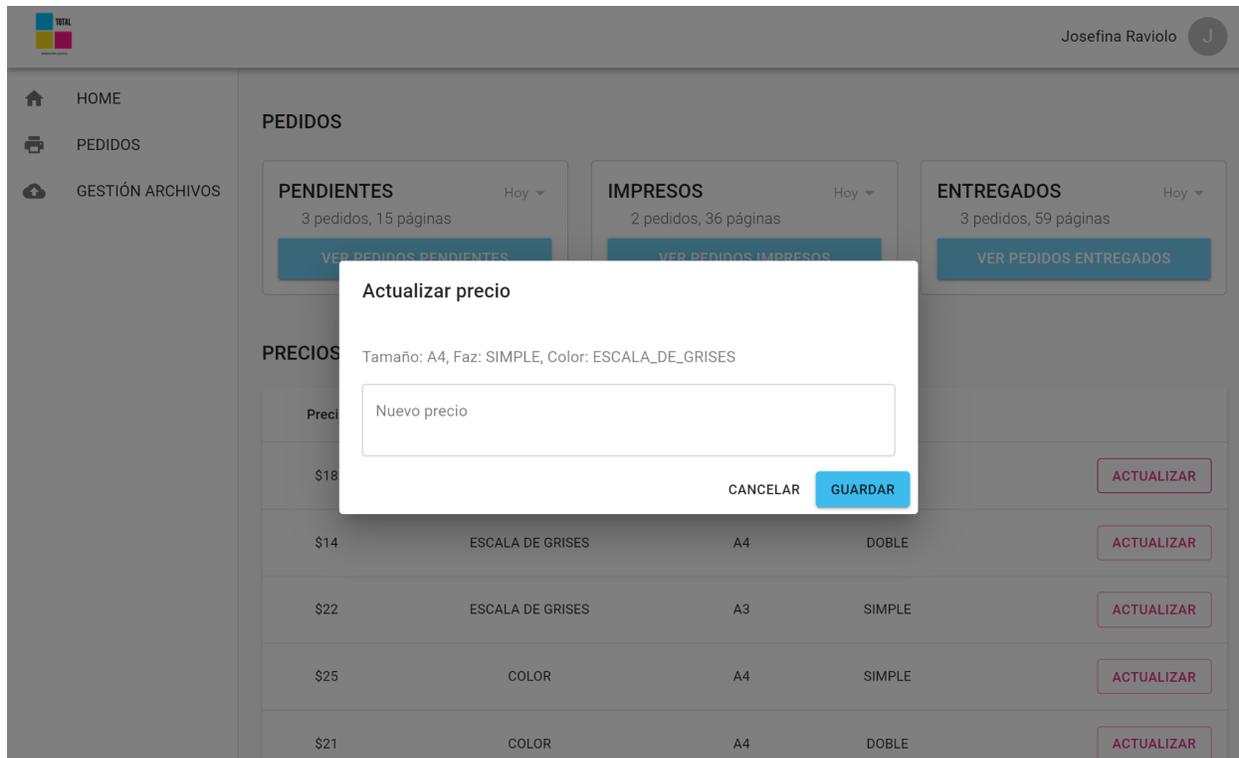


Figura 24: Dialog actualizar precios.

6.4.6 Ver listado de pedidos

Desde el menú “Pedidos” presente en la barra lateral se accede una interfaz en la que se muestra la lista completa de pedidos, ordenados inicialmente según la fecha de recepción del pedido. Sin embargo, todos los campos son filtrables y ordenables. En cada fila de la tabla se puede observar un botón con la leyenda “Ver pedido”, el cual nos llevará a ver el detalle del pedido seleccionado.

- HOME
- PEDIDOS
- GESTIÓN ARCHIVOS

PEDIDOS

ID	Nombre	Estado	Fecha estimada entrega	Email	
1	Julian	PENDIENTE	07/09/2021 a las 12:00	julian@gmail.com	VER PEDIDO
2	Alejandro Vola	ENTREGADO	07/09/2021 a las 10:00	aleeevola@gmail.com	VER PEDIDO
3	Alejandro Vola	IMPRESO	01/10/2021 a las 15:00	aleeevola@gmail.com	VER PEDIDO
8	Carlos Gomez	ENTREGADO	24/10/2021 a las 15:00	carlosgomez1@gmail.com	VER PEDIDO
10	Jose raviolo	IMPRESO	24/10/2021 a las 11:30	josee.r@live.com.ar	VER PEDIDO
13	Julieta Perez	ENTREGADO	27/10/2021 a las 09:00	perezJulieta95@gmail.com	VER PEDIDO
15	Jose raviolo	PENDIENTE	28/10/2021 a las 12:30	jraviolo@frsf.utn.edu.ar	VER PEDIDO
18	angelina	ENTREGADO	30/10/2021 a las 00:00	angelinaraviolo1@gmail.com	VER PEDIDO
21	Cristian	IMPRESO	03/11/2021 a las 12:15	cristianejemplo@gmail.com	VER PEDIDO

1-9 of 17 < >

Figura 25: Lista de pedidos.

6.4.6.1 Ver pedidos según estado

Desde el panel principal de administración se podía observar un resumen con un contador de pedidos según su estado y un botón con la leyenda “ver pedidos <estado>”. Dicho botón nos lleva a la funcionalidad de ver la lista de pedidos que están en el estado seleccionado.



- HOME
- PEDIDOS
- GESTIÓN ARCHIVOS

PEDIDOS PENDIENTES

ID	Nombre	Estado	Fecha estimada entrega	Email	
1	Julian	PENDIENTE	07/09/2021 a las 12:00	julian@gmail.com	VER PEDIDO
15	Jose raviolo	PENDIENTE	28/10/2021 a las 12:30	jraviolo@frsf.utn.edu.ar	VER PEDIDO
63	Rodolfo	PENDIENTE	24/11/2021 a las 09:38	ejemplo@gmail.com	VER PEDIDO

1 row selected

1-3 of 3 < >

Figura 26: Lista de pedidos pendientes.

6.4.7 Ver detalle de pedido

Tanto desde la lista de todos los pedidos como desde la lista de pedidos según su estado, se podrá acceder al detalle de cada pedido.

En esta interfaz se pueden ver los datos proporcionados por el cliente, es decir, su nombre y correo electrónico, junto con la hora programada para el retiro del pedido. A la derecha se observa un resumen del estado del pedido y si fue abonado o no. Según el estado en el que se encuentre el pedido, se mostrará un botón para registrar en el sistema el cambio de estado al siguiente estado permitido.

Más abajo se presenta la lista de archivos incluidos en el pedido en una tabla que detalla, además, las preferencias de impresión de cada uno. Por último, también se dispone de un botón para imprimir los archivos.



- HOME
- PEDIDOS
- GESTIÓN ARCHIVOS

PEDIDO #8

Detalles del cliente y pedido

Nombre	Carlos Gomez
Mail	carlosgomez1@gmail.com
Fecha programada de entrega	24/10/2021 a las 15:00

Pago #58
EXITOSOEstado pedido:
PENDIENTE**MARCAR IMPRESO**

ARCHIVOS

Archivo	Observaciones	Color	Tamaño	Faz	
Poliza Seguro 2021.pdf		COLOR	A4	SIMPLE	IMPRIMIR
IdeaProyecto-2020-Raviolo, Vola.pdf		ESCALA DE GRISES	A4	DOBLE	IMPRIMIR

Figura 27: Detalle de pedido.

6.4.8 Imprimir archivo

Al clicar en el botón presente en la tabla, se abre el cuadro de diálogo de impresión del navegador (Figura 28), donde se deben completar manualmente las preferencias de impresión, como es habitual en los centros de copiado.

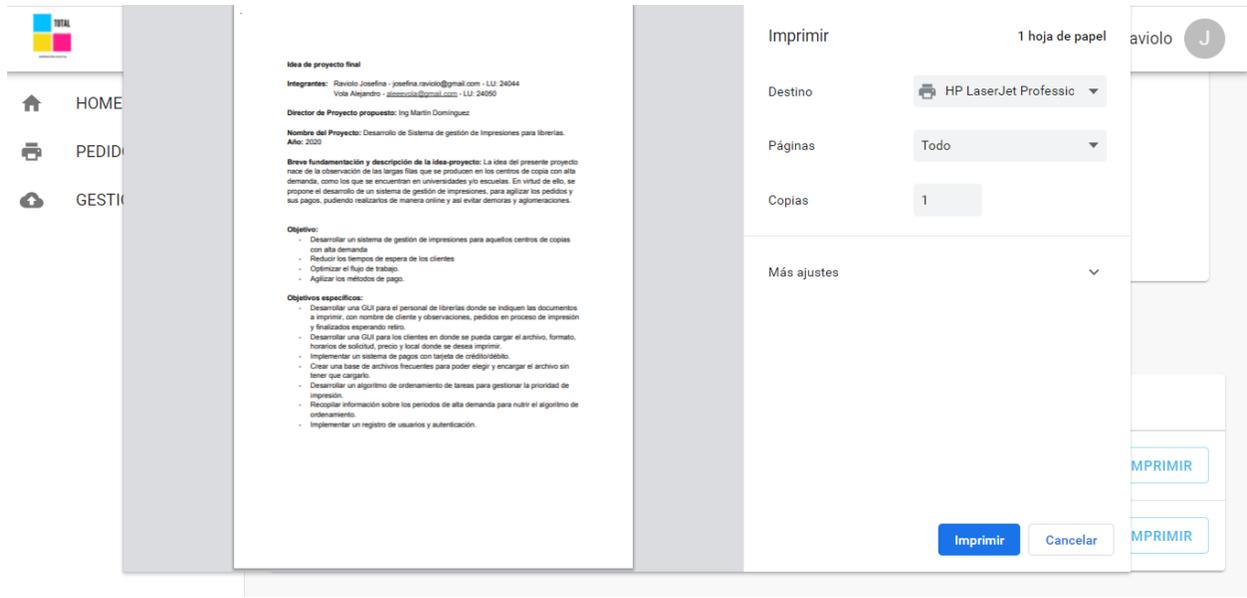


Figura 28: Imprimir archivo.

6.4.9 Cambiar estado del pedido

El pedido que se utilizó para la demostración fue el #8, el cual como se pudo observar en la Figura 27, se encontraba en estado “PENDIENTE”. Esto significa que el siguiente estado sólo puede ser “impreso”. Al presionar el botón “marcar impreso”, se puede ver en la captura que se presenta en la Figura 29 que el estado del pedido ahora es “IMPRESO” y que el botón ahora permite marcar el pedido como “entregado”, que es el siguiente estado permitido en el circuito.



- HOME
- PEDIDOS
- GESTIÓN ARCHIVOS

PEDIDO #8

Detalles del cliente y pedido

Nombre	Carlos Gomez
Mail	carlosgomez1@gmail.com
Fecha programada de entrega	24/10/2021 a las 15:00

Pago #58
EXITOSOEstado pedido:
IMPRESO**MARCAR ENTREGADO**

ARCHIVOS

Archivo	Observaciones	Color	Tamaño	Faz	
Poliza Seguro 2021.pdf		COLOR	A4	SIMPLE	IMPRIMIR
IdeaProyecto-2020-Raviolo, Vola.pdf		ESCALA DE GRISES	A4	DOBLE	IMPRIMIR

Figura 29: Detalle de pedido con nuevo estado.

6.4.10 Lista archivos frecuentes

La lista de archivos frecuentes es accesible desde el menú presente en la barra lateral “gestión archivos”. Al presionarlo, se mostrará la interfaz que muestra la Figura 30.



- HOME
- PEDIDOS
- GESTIÓN ARCHIVOS

ARCHIVOS FRECUENTES

SUBIR ARCHIVO

Lista de archivos frecuentes

AnexoDiagramasDeGantt-2021-Raviolo, Vola.pdf

ELIMINAR

IdeaProyecto-2020-Raviolo, Vola.pdf

ELIMINAR

ENTRADAS DON OSVALDO.pdf

ELIMINAR

Diagramas de Gantt - Sheet1.pdf

ELIMINAR

Apuntes - 3 - Números Complejos.pdf

ELIMINAR

Apuntes - 5 - Función de Transferencia.pdf

ELIMINAR

Diapositivas - 2 - Linealización de SDs.pdf

ELIMINAR

Diapositivas - 1 - Modelado de Sistemas.pdf

ELIMINAR

Figura 30: Lista de archivos frecuentes

La Figura 30 muestra la lista, junto con la opción de eliminar los archivos, mientras que en la parte superior derecha del panel se puede ver un botón para subir archivos, lo que nos lleva a la última funcionalidad implementada para esta primera versión del sistema.

6.4.11 Subir archivo frecuente

Por último, se puede ver que presionando el botón “subir archivo” se abrirá un cuadro de diálogo del navegador para seleccionar un archivo (Figura 31). Una vez seleccionado el mismo, se mostrará el cuadro de diálogo que se presenta en la Figura 32, en el que se detalla el nombre del archivo y la cantidad de páginas

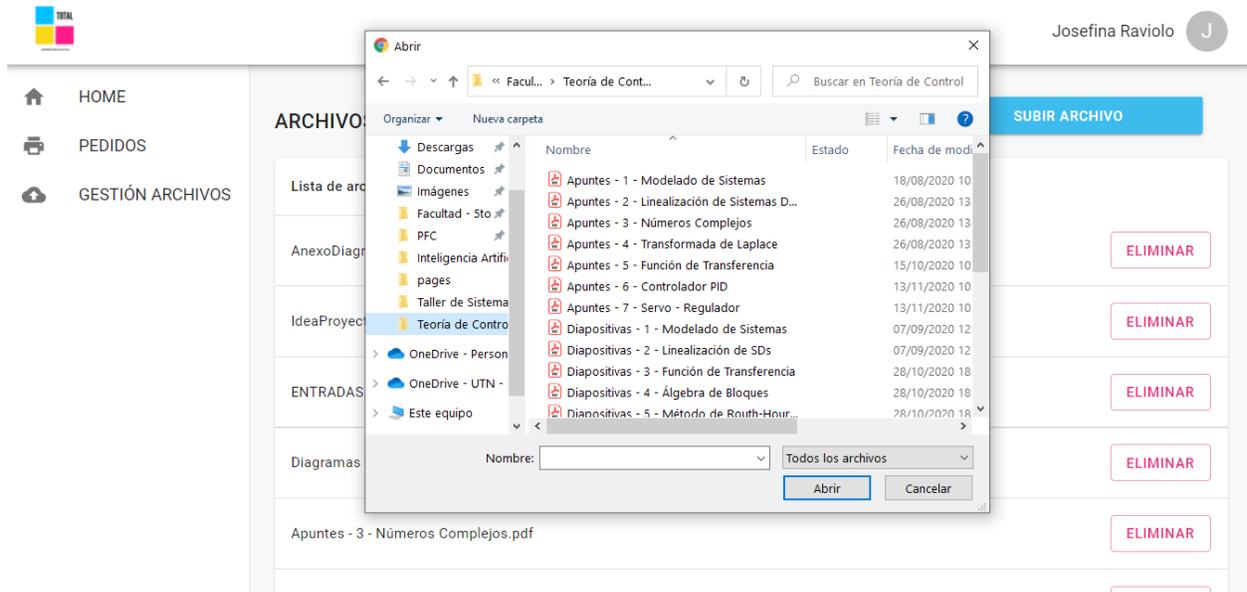


Figura 31: Diálogo para seleccionar archivo frecuente.

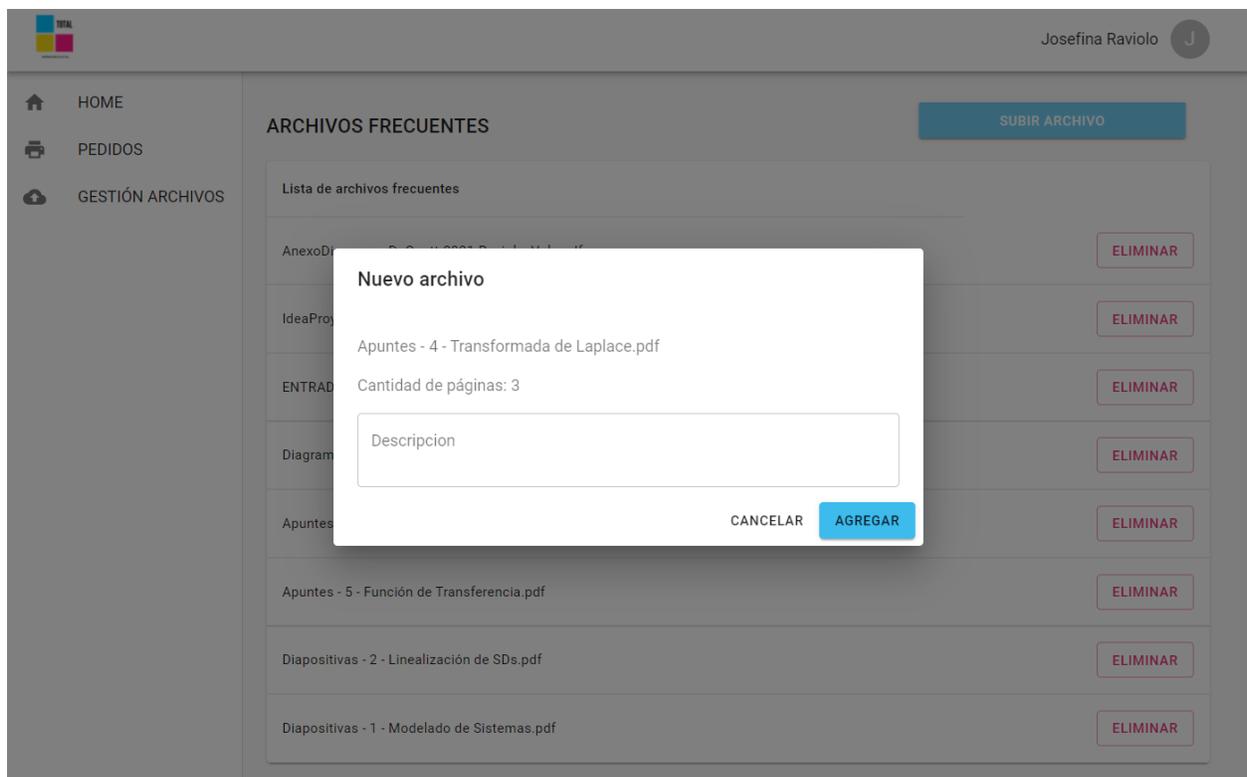


Figura 32: Diálogo para subir archivo frecuente.

7. Aseguramiento de la calidad

Las pruebas de software son una parte muy importante en cualquier proceso de desarrollo. Las mismas ayudan a verificar el correcto funcionamiento del producto desarrollado y que el mismo cumpla con los criterios de aceptación del cliente.

Para el aseguramiento de la calidad del software desarrollado, se implementaron tanto pruebas automáticas como pruebas manuales. En el presente apartado se explican y ejemplifican algunas de ellas, separándolas en pruebas del backend y del frontend.

Existen también otros tipos de pruebas, tales como pruebas de carga, esfuerzo, performance, entre otras, pero dada la magnitud del sistema, el equipo decidió no incluirlas.

7.1 Backend

Para las pruebas del backend se realizaron tanto pruebas manuales como automáticas. Éstas últimas son una gran herramienta para que el ciclo de desarrollo general sea más rápido, ya que ayudan a descubrir y corregir fallas, así como reducir el tiempo de pruebas cuando se producen cambios o evoluciones en el código.

En el marco de las pruebas o test realizados en el proyecto, se pueden destacar pruebas unitarias, pruebas de integración y pruebas del servicio Restful completo, las cuales se explican en las siguientes secciones.

7.1.1 Pruebas automatizadas

En el proceso de desarrollo se automatizaron tanto pruebas unitarias como pruebas de integración.

Las pruebas unitarias son responsables de probar una unidad funcional y aislada de código de una aplicación de software, para determinar su corrección y comportamiento esperado.

Por su parte, las pruebas de integración son responsables de probar la integración/comunicación entre diferentes unidades funcionales del software.

Para la realización de las pruebas se utilizó JUnit, un framework open source para la automatización de pruebas en Java. Las mismas consistieron en realizar tests en los métodos más críticos de las clases *Service*.

A continuación se listan los tests resultantes, y se brinda un ejemplo en la Figura 33.

7.1.1.1 Archivos Frecuentes

- `ObtenerArchivosFreguntes`: devuelve una lista de los archivos frecuentes.

7.1.1.2 Archivos

- `ObtenerNumeroDePaginas`: dado un `Archivo` retorna la cantidad de páginas.
- `ObtenerPrecioArchivo`: dado un determinado número de páginas y las preferencias de impresión de un archivo (faz, color y tamaño de hoja), retorna el precio en una variable `float`.
- `SubirArchivoSinPedido`: dado un archivo, con su preferencias de impresión (faz, color y tamaño de hoja) y sin referenciar a un pedido, crea un nuevo pedido y guarda el archivo.

7.1.1.4 File Storage

- `GuardarArchivo`: dado un archivo y un identificador numérico guarda el mismo en una carpeta y retorna su identificador en una variable del tipo `string`.

7.1.1.5 Pagos

- `ObtenerLinkDePago`: dado un identificador de un pedido retorna una URL de `mercadoPago` y genera un pago con su método de pago como `mercadoPago`.

- PagoEfectivo: Dado el id del pedido cambia sus estado a pendiente y genera un pago con su método de pago como efectivo.

7.1.1.6 Pedido

- ObtenerPedido: dado el identificador del pedido retorna el correspondiente.
- ObtenerEstadoPedido: dado el identificador del pedido retorna el estado actual del mismo.

7.1.1.7 Precio

- ObtenerTodosLosPrecios: retorna una lista con las características y precios correspondientes a una página de un archivo.
- ObtenerPrecio: Dada las preferencias de impresión de un archivo: faz, color y tamaño de hoja, retorna el precio que le corresponde a una página.

```
1 @Test
2 public void ObtenerNumeroDePaginas() throws IOException {
3     Path resourceDirectory = Paths.get("src", "test", "resources", "prueba.pdf");
4     String absolutePath = resourceDirectory.toFile().getAbsolutePath();
5
6     MultipartFile multipartFile = new
7     MockMultipartFile("prueba.pdf", "prueba.pdf", "application/pdf", new FileInputStream(new
8     File(absolutePath)));
9
10    int numeroPaginas = _archivosServices.getNumeroPaginas(multipartFile);
11    assertNotNull(multipartFile);
12    assertEquals(2, numeroPaginas);
13 }
14 @Test
15 public void ObtenerPrecioArchivo(){
16     int numeroPaginas=3;
17     float valorEsperado = PRECIO_SIMPLE_A4_COLOR*numeroPaginas;
18     assertEquals(valorEsperado, _archivosServices.getPrecio(numeroPaginas,
19     TipoImpresion.DOUBLE, TamanoHoja.A4, Color.COLOR),0.001);
20 }
```

Figura 33: Ejemplo tests “Obtener número páginas” y “obtener precio archivo”.

7.1.2 Pruebas API Rest

Como fue mencionado previamente, se utilizó la herramienta *Swagger* para probar el servicio Restful desarrollado.

El cliente de *Swagger* permite visualizar todos los controladores y endpoints disponibles (Figura 34), así como probar su correcto funcionamiento.

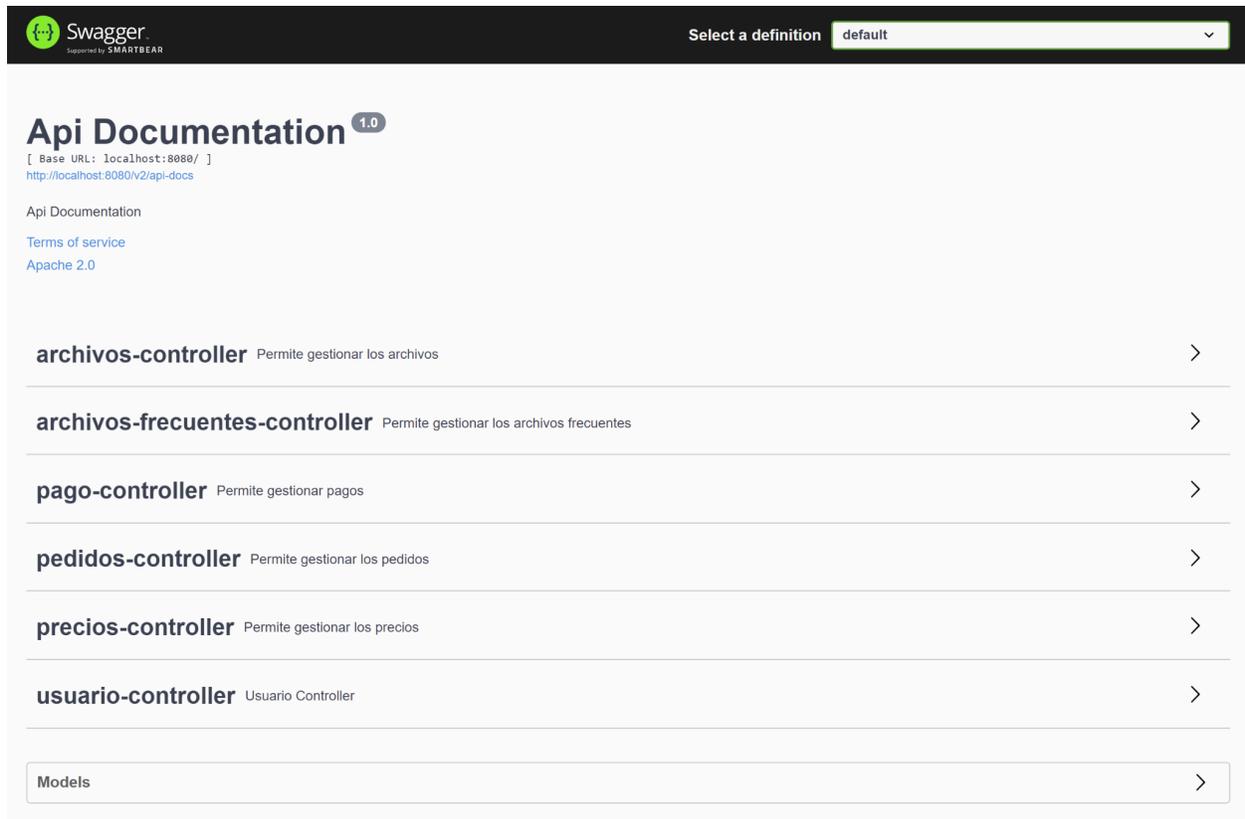


Figura 34: Cliente swagger.

Como se puede observar en la figura 35, al seleccionar un endpoint se muestran los parámetros de entrada, un ejemplo de respuesta correcta y también un botón para ejecutar el mismo.

archivos-controller Permite gestionar los archivos >

archivos-frecuentes-controller Permite gestionar los archivos frecuentes >

pago-controller Permite gestionar pagos >

pedidos-controller Permite gestionar los pedidos >

GET /pedidos Obtener todos los pedidos..

GET /pedidos/{idPedido} Obtener un pedido

Parameters Try it out

Name	Description
idPedido * required integer(\$int32) (path)	idPedido

idPedido - idPedido

Responses Response content type */*

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

Example Value | Model

```

{
  "archivos": [
    {
      "color": "COLOR",
      "fechaBaja": "2021-12-01",
      "fechaIngreso": "2021-12-01",
      "idArchivo": 0,
      "nombre": "string",
      "numeroPaginas": 0,
      "observaciones": "string",
      "precio": 0,
      "tamañoHoja": "A1",
      "tipoImpresion": "DOBLE",
      "token": "string"
    }
  ],
  "cantidadArchivos": 0,
  "estado": "CANCELADO",
  "fechaEstimadaEntrega": "2021-12-01T21:33:08.560Z",
  "idPedido": 0,
  "pago": "PAGADO",
  "usuario": {
    "activo": true,
    "email": "string",

```

PATCH /pedidos/actualizarEstado Actualizar estado del pedido

GET /pedidos/cantidadPorEstado Obtiene cantidad de pedidos por estado en fecha

GET /pedidos/estado Obtener el estado de un pedido

Figura 35: Ejemplo vista endpoint swagger.

7.2 Frontend

En cuanto al aseguramiento de la calidad del frontend de la aplicación, los componentes utilizados en Nextjs se probaron de manera manual por el equipo de desarrollo.

Por otra parte, las interfaces de usuario fueron validadas por el cliente según la metodología que se explica en el siguiente apartado. Cabe aclarar que las mismas se diseñaron y desarrollaron manteniendo una estética minimalista, consistente y moderna, respetando un estilo para cada rol de usuario y los lineamientos de marca de la empresa “Total impresión digital”.

7.2.1 Validación del cliente

A fin de validar las interfaces de usuario de la aplicación, inicialmente se diseñaron mockups o maquetas, para mostrar al cliente las interfaces tentativas y poder obtener el feedback del mismo.

Luego, una vez desarrolladas las interfaces finales, se realizaron reuniones virtuales con el cliente a fin de mostrar tanto las interfaces como el flujo de trabajo dentro de la aplicación.

En este contexto, también se realizaron pruebas alfa. Las pruebas alfa *“son pruebas de software realizadas cuando el sistema está en desarrollo y cuyo objetivo es asegurar que lo que estamos desarrollando es probablemente correcto y útil para el cliente”*. Gracias a ello se logró la validación del dueño y los empleados de la empresa.

8. Extensibilidad

Como se mencionó anteriormente, el proyecto contempla el desarrollo de un producto mínimo viable. En caso de que el mismo tenga buena aceptación tanto por los clientes como por los usuarios administradores, se prevé la extensión del sistema a una segunda versión que incluya las siguientes funcionalidades:

8.1 Refactorización de clases de archivos

Como se mencionó previamente, las clases Archivo y ArchivoFrecuente deberían unificarse ya que poseen la mayoría de sus atributos en común.

8.2 Crear cuenta usuarios clientes

Si bien ya se posee una entidad “Usuario”, no están implementadas las funcionalidades para registrarse, iniciar y cerrar sesión y demás funcionalidades asociadas a la cuenta, como ver todos los pedidos realizados, entre otras.

8.3 Estadísticas de administración

En cuanto al rol de administración, se pueden implementar estadísticas sobre la cantidad de pedidos recibidos, el valor de los mismos, análisis de horarios de mayor demanda, entre otras.

8.4 Algoritmos de ordenamiento de tareas

Para los usuarios administradores, se puede implementar un algoritmo de ordenamiento de tareas que optimice el tiempo para terminar la mayor cantidad de pedidos en menor tiempo

8.5 Categorías de archivos frecuentes

En los requerimientos iniciales surgió la idea de poder asignar categorías y subcategorías a los archivos frecuentes para facilitar la búsqueda de los mismos en caso de no saber el nombre con el que se encuentran cargados en el sistema.

8.6 Crear cuenta usuarios clientes con diferentes roles

En las reuniones con el cliente se mencionó la posibilidad de crear aunque sea dos tipos de usuarios clientes. Esto surge de la frecuencia con que los profesores de distintas escuelas dejan material para sus alumnos, por ende los roles quedarían definidos por uno de “Profesor” y otro de “Alumno” o “Cliente convencional”. El primero de éstos podría subir los archivos frecuentes y dejarlo disponible para que lo incluyan en sus pedidos sus Alumnos.

8.7 Parametrizar los horarios

En futuras iteraciones se podría agregar una sección al menú del administrador para configurar los intervalos en los cuales la empresa atiende al público. Logrando así que éstos se actualicen en todo el sistema y los clientes siempre realicen pedidos en el horario correcto. Actualmente para cambiar los horarios permitidos en el proceso de crear un pedido hay que modificar variables en el frontend, lo que implica un deploy a producción.

8.8 Crear un sistema genérico

Adaptar el desarrollo a un sistema genérico para múltiples empresas del mismo rubro sería una oportunidad de negocio interesante.

9. Aportes y conclusiones

9.1 Desde el punto de vista de la empresa

El producto desarrollado, si bien es una versión preliminar, presenta diversas mejoras y soluciones en el proceso de negocio tanto para la empresa como para los clientes de la misma.

9.1.1 Para la empresa

Previamente al desarrollo de la aplicación, la empresa recibía pedidos por diversos medios digitales, tales como instagram, correo electrónico y whatsapp. Si bien el tener más de un canal es una ventaja para poder captar mayor cantidad de clientes, esto presenta una serie de inconvenientes, ya que no se centralizan a nivel empresa los pedidos, lo cual dificulta la gestión de los mismos. El dueño de la empresa planteó esta dificultad ya que le preocupaba que el servicio a sus clientes fuera deficiente, así como el aprovechamiento de los recursos, desde impresoras hasta personal del centro de impresiones.

Ahora, con la llegada del nuevo software, se presenta la posibilidad de centralizar los pedidos, agilizando la forma de recepción y estandarizando las especificaciones de los mismos, pudiendo percibir el pago de las órdenes por adelantado, lo que permitirá imprimir los pedidos cuando son recibidos, optimizando tiempos muertos sin arriesgar a tener pérdidas por haber impreso pedidos que nunca serán retirados.

La nueva aplicación representa también una modernización del negocio, en un contexto en el que es de suma importancia para las pequeñas y medianas empresas acceder a tecnologías y soluciones profesionales, ya que se traduce en una mejora de oportunidades de crecimiento.

Por último, se espera que también los colaboradores de la empresa perciban una mejora en su trabajo diario al encontrarse en un entorno que avanza, sintiéndose

más cómodos y productivos en sus tareas al poder ordenar las mismas de mejor manera.

9.1.2 Para los clientes de la empresa

Desde el punto de vista de los clientes de la empresa también se espera una buena aceptación de parte de los mismos, ya que brinda una solución rápida, práctica y confiable para sus pedidos de impresión, desde la generación hasta el pago.

Esto soluciona a la vez un problema al que no se hizo mención previamente en el informe, ya que es muy común en el contexto actual de Argentina. A lo largo de los últimos años, se fueron quitando de circulación los billetes de menor denominación, es decir, los billetes de 2 pesos y los de 5 pesos. Sin ahondar en los detalles, motivos y consecuencias de esas decisiones, es común que en los negocios que se manejan precios relativamente pequeños, o donde los montos de los pagos de los clientes no sean múltiplos de 10, se terminen redondeando a veces en favor del cliente y a veces en favor de la empresa, o que se implemente un sistema de “vales” para utilizar en otra compra que se realice en la empresa. En este contexto, poder realizar los pagos mediante la plataforma Mercado Pago, brinda a ambas partes la tranquilidad de que se respetará el precio de la orden, sin incurrir en el sistema de vales que resulta muchas veces poco conveniente para los clientes.

La aplicación también eficientiza el retiro de los pedidos, pudiendo evitarse las filas de espera, gracias a la mejoras que introduce en la planificación de la empresa y a sus sistema de envío automático de correos electrónicos que informan al cliente cuando su pedido está listo para ser retirado.

9.2 Desde el punto de vista del producto

Por las características del negocio de impresiones en general en la ciudad de Santa Fe y las características del producto desarrollado, se cree que se puede adaptar y hacer extensibles a otras empresas del rubro, en especial aquellas que se encuentran dentro o en cercanías de escuelas y universidades.

Como se describió en secciones anteriores, las buenas prácticas aplicadas en el desarrollo de la aplicación, hacen de éste un producto fácilmente adaptable a distintas preferencias de otros potenciales clientes.

El sistema desarrollado en el marco de este proyecto final de carrera, brinda una solución tanto a empresas como a sus clientes finales, dando soporte en todo el proceso del negocio, desde la generación de pedidos de impresión, pasando por el pago, hasta el retiro del pedido, y facilitando las tareas de comunicación entre la empresa y sus clientes.

Por último, dada la magnitud de la aplicación, el equipo cree que es una oportunidad para las pymes locales de modernizar y eficientar sus recursos de una manera accesible ya que se trataría de una adaptación a la aplicación desarrollada para “Total Impresión Digital”.

9.3 Desde el punto de vista profesional

Como primera conclusión, se notó que la estimación de esfuerzos y velocidad de desarrollo es muy difícil de determinar acertadamente cuando no se poseen los suficientes conocimientos de las tecnologías a utilizar.

Más allá de esto, se quiere destacar una serie de aportes que representó la realización de este proyecto principalmente desde un punto de vista profesional, pero también desde el punto de vista personal, ya que se sabe que en la actualidad son cuestiones muy relacionadas.

El primer aporte sustancial que se puede sacar de este proyecto es el conocimiento en una de las tecnologías de frontend más utilizadas y requeridas por las empresas actualmente como es NextJs - React, lo cual fue uno de los motivos para seleccionarla en la realización de las interfaces de usuario de la aplicación.

En esta línea también se destaca el uso e integración de todas las tecnologías utilizadas, puesta en funcionamiento y configuración de todo el entorno, el

conocimiento adquirido en el framework Spring Boot, lo que es un gran complemento a nuestra formación en Java como desarrolladores de backend.

Asimismo, el tener que definir como equipo la metodología de desarrollo, tratar de ajustarse lo máximo posible al plan y a la metodología, decidir las tecnologías a utilizar en un mundo en constante cambio y evolución, con infinidad de opciones disponibles que todas tienen aspectos que se adaptan al proyecto, el sortear los obstáculos que se presentaron, la comunicación con el cliente y la autogestión del equipo fueron aspectos relevantes que producen un crecimiento profesional significativo en los participantes.

Por último, pero no por ello menos importante, se destaca la participación, la cooperación, el compromiso y respeto de ambos compañeros de equipo. Priorizando siempre las relaciones humanas, la amistad existente y el bienestar de los integrantes durante el desarrollo del proyecto. Esto representa no sólo gran aporte en lo personal sino también en lo profesional, ya que el equipo está convencido que las habilidades blandas, o “soft skills” en inglés, son también un punto muy importante en la formación profesional tanto para cargos de mandos como operativos, porque mejoran enormemente el ambiente de trabajo y, en consecuencia, la productividad y sinergia del equipo.

10. Referencias

Pressman, Roger S. "Ingeniería de Software: Un Enfoque Práctico", 7ma Edición. Mc Graw-Hill. 2010.

Sommerville, Ian. "Software Engineering". 9th Edition. Pearson Education, 2011.

Apuntes de cátedra.

Wikipedia - Teorema CAP - wikipedia.org

AWS: ¿Qué son los microservicios? - aws.amazon.com

Rock Content: API REST ¿Qué es y cómo utilizar ese recurso? - rockcontent.com

Arteco-consulting: Por qué debes usar Spring Boot - arteco-consulting.com

Rock Content: GitHub ¿Qué es y por qué es tan importante en un negocio online? - rockcontent.com

Auth0: Access management - Handle users identity - auth0.com

Infobae: El guardián de las passwords. La historia de Auth0 - infobae.com

Wikipedia: Swagger software - wikipedia.org

Ramon Abad: ¿Qué es Swagger? - ramonabadyapuntonet.org

Genbeta: Eclipse IDE - genbeta.com

ANEXO A - Historias de usuario

Nombre de la historia: ABMC de usuarios de administración (centro de copiado)	
Descripción	
<p>Desde la cuenta de administrador se debe poder dar de alta usuarios nuevos del centro de copiado. También se debe poder elegir qué permisos tiene cada usuario, tales como ver los pedidos, imprimirlos, crear nuevos usuarios.</p> <p>En la pantalla de gestión de usuarios debe mostrarse una vista previa de todos los usuarios dados de alta y permitir modificarlos o eliminarlos.</p> <p>Eliminar usuarios de empleados que ya no trabajan.</p>	
Tareas	
1.1	Interfaz de gestión de usuarios
1.2	Interfaz de alta/modificación de usuario
1.3	Modificar usuario en la base de datos
1.4	Eliminar usuario de la base de datos

Nombre de la historia: Inicio/cierre de sesión de usuarios de administración	
Descripción	
<p>Debe haber una pantalla de inicio de sesión al iniciar el sistema en la cual los usuarios de administración puedan loguearse con su usuario y contraseña.</p> <p>Luego de iniciar sesión se tiene que mostrar el dashboard de administración.</p> <p>En cualquier interfaz de administración del sistema se debe poder cerrar sesión.</p>	
Tareas	
2.1	Autenticación de usuarios de administración
2.2	Cierre de sesión

Nombre de la historia: ABM de usuarios	
Descripción	
Los clientes deben poder registrarse en nuestro sitio para realizar pedidos y seguimiento de sus pedidos. Deben tener nombre, apellido y correo electrónico.	
Tareas	
3.1	Interfaz registro de usuario
3.2	Interfaz de login de usuario
3.3	Persistencia de usuarios en la base de datos.
3.4	Backend de ABM de usuarios.

Nombre de la historia: ABMC de pedidos	
Descripción	
Cada usuario/cliente puede realizar un pedido de impresión de archivos, cada uno de ellos puede solicitar impresión doble faz o simple, a color o blanco y negro. Cada pedido tendrá una fecha y hora de retiro por la sucursal ingresada por el usuario, esto debe ser tenido en cuenta al momento de imprimir. El usuario puede seleccionar archivos de su propio ordenador o archivos frecuentes que se encuentran cargados en nuestro sistema. Todos los pedidos antes de ser realizados deben estar pagados.	
Tareas	
4.1	Interfaz de realizar pedido
4.2	Interfaz archivos disponibles
4.3	Backend ABMC de pedidos
4.4	Persistencia de pedidos en base de datos
4.5	Interfaz de estado de pedido

Nombre de la historia: AMB de archivos	
Descripción	
Los archivos que pueden ser subidos deben tener formato “.doc” ó “.pdf”, tamaño de hoja A3 o A4. Estos se pueden subir desde la realización de pedidos	
Tareas	
5.1	Persistencia de archivos en la base de datos
5.2	Persistencia de archivos en el file system
5.3	Front subir un archivo en pedido
5.4	Backend amb archivos

Nombre de la historia: AMBC de archivos frecuentes	
Descripción	
Los archivos que pueden ser subidos deben tener formato “.doc” ó “.pdf”, tamaño de hoja A3 o A4. Estos pueden ser subidos por un usuario de administración que tenga permiso desde el panel subir archivos frecuentes o desde la interfaz de gestión de archivos frecuentes. Se debe poder eliminar archivos y modificar el nombre/descripción de los existentes.	
Tareas	
6.1	Interfaz subir archivos frecuentes
6.2	Persistencia de archivos en la base de datos
6.3	Modificar archivos de la base de datos
6.4	Modificar ubicación de los archivos en el file system
6.5	Eliminar archivos de la base de datos

Nombre de la historia: Integración del procesador de pagos	
Descripción	
Los pedidos deben ser pagados con la aplicación de Mercado Pago	
Tareas	
7.1	Implementación de API Mercado Pago

7.2	Configuración y linkeo de cuenta del propietario
-----	--

Nombre de la historia: Gestor de alertas por correo electrónico	
Descripción	
<p>Cuando un usuario nuevo se registra debe recibir un correo de confirmación de cuenta. Cada vez que un usuario realiza el pedido recibe una notificación con un link donde puede revisar el estado de su pedido.</p> <p>Cuando un pedido está impreso por completo el usuario recibe un email para poder retirarlo por el local.</p>	
Tareas	
8.1	Configuración de servicio de correo electrónico
8.2	Configuración de notificaciones
8.3	Formato y contenido de correo electrónico

Nombre de la historia: Gestionar pedidos	
Descripción	
<p>El usuario administrador deberá ver en el dashboard todos los pedidos pendientes, de cada uno se debe mostrar la fecha y hora límite de entrega, la cantidad de páginas, observaciones y si el mismo fue pagado o no.</p> <p>En otra interfaz se debe poder ver todos los pedidos recibidos junto con el estado de cada uno y qué usuario los imprimió en caso de estar impreso.</p> <p>Cuando el pedido es ingresado tiene un estado “pendiente”, cuando es impreso a “impreso” y cuando es cancelado a “cancelado”.</p>	
Tareas	
9.1	Interfaz de pedidos pendientes
9.2	Interfaz de pedidos recibidos
9.3	Interfaz de impresión
9.4	Imprimir documento
9.5	Actualizar estado pedido

Nombre de la historia: Gestor de impresoras	
Descripción	
El usuario administrador podrá agregar, quitar o modificar impresoras que se encuentran disponibles en el local.	
Tareas	
10.1	Interfaz gestor de impresoras.
10.2	Configuración de impresoras.

Nombre de la historia: Interfaz principal	
Descripción	
La aplicación web deberá tener una página principal donde se indiquen contactos de la empresa y un tutorial de como realizar pedidos por la misma.	
Tareas	
11.1	Interfaz principal
11.2	Imágenes de guía

Nombre de la historia: Interfaz inicial administrador	
Descripción	
El administrador deberá tener una pantalla con los accesos directos a las diferentes acciones que puede realizar.	
Tareas	
12.1	Interfaz administrador