



Revista Micaela

ISSN: 2955-8646 (en línea) / 2709-8990 (Impresa)
Universidad Nacional Micaela Bastidas de Apurímac
Vice Rectorado de Investigación – Perú

Vol. 5 Num.2 (2024) - Publicado: 22/03/24

DOI: <https://doi.org/10.57166/micaela.v5.n2.2024>

Páginas: 1 - 8

Recibido 20/08/2024; Aceptado 04/09/2024

<https://doi.org/10.57166/micaela.v5.n2.2024.145>

Autores:

1. **ORCID iD** <https://orcid.org/0009-0004-7890-5974>
Oscar Choquehuallpa Hurtado, Egresado de la Universidad Nacional Micaela Bastidas de Apurímac – Pe
161139@unamba.edu.pe
2. **ORCID iD** <https://orcid.org/0009-0007-7564-4272>
Jonathan Bustinza Mendoza, Egresado de la Universidad Nacional Micaela Bastidas de Apurímac Perú– Pe
162134@unamba.edu.pe
3. **ORCID iD** <https://orcid.org/0000-0002-2552-5669> Mario Aquino Cruz, Docente en la Universidad Nacional Micaela Bastidas de Apurímac Perú– Pe
maquino@unamba.edu.pe

Desarrollo de una aplicación de escritorio para la impresión de tickets en los sistemas de punto de venta de la empresa PUYU

Development of a desktop application for printing tickets in the point of sale systems of the PUYU Company

Oscar Choquehuallpa-Hurtado¹, Jonathan Bustinza-Mendoza²
Mario Aquino-Cruz³

Resumen. Este trabajo de investigación presenta la implementación de una aplicación de escritorio para la impresión de tickets, el sistema de punto de venta consume el API de la aplicación para imprimir varios tipos de documentos en diferentes tipos de ticketeras. Es una investigación de tipo aplicada y de alcance descriptivo, la implementación de la aplicación se realizó utilizando la metodología ágil Scrum. Las pruebas se realizaron en la pizzería Il Gatto ubicada en la provincia de Andahuaylas; asimismo se aplicó un cuestionario a 4 personas que constaba de 8 preguntas, en el cual se obtuvo un puntaje general de 4.6 en una escala del 1 al 5, lo cual indica que los usuarios están satisfechos con la usabilidad de la aplicación.

Palabras Clave: Aplicación de escritorio, impresión de tickets, metodología Scrum, punto de venta.

Abstract. This research work presents the implementation of a desktop application for printing tickets, the point of sale system consumes the application's API to print various types of documents in different types of ticket holders. It is an applied research of descriptive scope; the implementation of the application was carried out using the agile Scrum methodology. The tests were carried out at the Il Gatto pizzeria located in the province of Andahuaylas; Likewise, a questionnaire was applied to 4 people that consisted of 8 questions, in which a general score of 4.6 was obtained on a scale of 1 to 5, which indicates that users are satisfied with the usability of the application.

Keywords: Desktop application, ticket printing, Scrum methodology, point of sale.

1 Introducción

En el contexto del presente artículo se define el sistema de punto de venta (POS por sus siglas en inglés) como el conjunto de hardware y software diseñado para optimizar los procesos de negocio de atención al cliente y ventas. Los componentes de un POS pueden variar, pero típicamente están compuestos por un software especializado, un escáner de códigos de barras, una impresora de tickets, una gaveta de dinero y una computadora con conexión a internet [1].

Un ticket es un comprobante de pago que emiten las máquinas registradoras cuando se adquieren bienes o servicios como consumidor o usuario final, y sirve para propósito personal [2], en términos simples un ticket de impresión es un documento generado por una impresora térmica o similar, utilizado para registrar transacciones comerciales, tales como boletas, comandas, recibos, precuentas o cualquier otro documento con información sobre algún proceso de venta, los tickets representan los movimientos de las ventas diarias en un negocio.

En este contexto PUYU es una empresa especializada en brindar servicios en la nube “Software como Servicio”, Actualmente la empresa cuenta con 3 sistemas de punto de venta: YuBiz, YuRes y YuBus, Todos ellos orientados a diferentes tipos de negocio como puntos de venta de negocios en general, restaurantes hasta empresas de transporte [3]. Adicionalmente a ello los tres sistemas de punto de venta (POS) integran un servicio de facturación electrónica desarrollada por la misma empresa, gracias a ello cada sistema puede emitir documentos electrónicos, estos documentos pueden ser entregados al cliente en formato electrónico, pero también es posible entregarlos en físico en formatos de tickets de impresión como boletas, facturas, notas de venta, etc.

Para la impresión de los tickets la empresa Puyu desarrolló un pequeño programa que ayuda con estas tareas denominado “FastPrinter” y que se incluyen en la instalación de dos de los sistemas de punto de venta que provee la empresa, YuBus y YuRes. YuBus es una solución de software orientado a empresas de transporte y YuRes orientado a bufets, puntos de comida rápida, restaurantes en general, etc. Tanto YuBus como YuRes utilizan FastPrinter para la gestión y diseño de impresión de los documentos electrónicos como boletas, facturas, notas de venta, guías de remisión, etc. Adicionalmente a ello YuRes lo utiliza para imprimir otros tipos de documentos como guías de entrega, comandas, precuentas, delivery, etc. FastPrinter no se limita al diseño de los documentos de impresión, sino que también, a gestionar los distintos tipos de conexión que existen con las ticketeras, tales como: conexión vía Ethernet, mediante protocolo Samba, puertos seriales, puertos virtuales, mediante USB, etc. Esta última característica es muy utilizada por YuRes, ya que los negocios de la mayoría de sus clientes tienen varias ticketeras con distintos tipos de conexión según las necesidades del negocio.

FastPrinter está desarrollada con el lenguaje de programación PHP, y se despliega de forma artesanal instalando programas de terceros como XAMPP para su correcto funcionamiento en la computadora de los clientes. PHP es un lenguaje orientado a desarrollar aplicaciones del lado del servidor y XAMPP es un conjunto de herramientas para el desarrollo web que incluye PHP [4]. Instalar XAMPP en la PC de los clientes es algo contra intuitivo ya que está pensado para el desarrollo web y no como algo que usarían los usuarios finales, sin embargo, FastPrinter depende de XAMPP para funcionar, lo que genera complicaciones innecesarias en la instalación de los sistemas de punto de venta, incluso la instalación y configuración de este paquete de herramientas de desarrollo varía según el sistema operativo. En el caso de YuRes, actualmente existen clientes que utilizan sistemas operativos como Windows, Linux y Mac, y por lo tanto FastPrinter no es una solución multiplataforma.

Asimismo, FastPrinter no tiene un mecanismo de actualizaciones automáticas. En caso de tener que desplegar nuevas funcionalidades o corregir algunos posibles errores, se tiene que acceder a la PC del usuario final para poder actualizar manualmente el servicio de impresión, para lo cual utiliza GIT como herramienta para traer los nuevos cambios. GIT es una herramienta para el control de versiones de un software [5], y su uso en los usuarios finales es atípico, ya que está pensada para desarrolladores de software. Sin mencionar que esto implica manejar las versiones de cada cliente manualmente, este problema complica demasiado el proceso si son muchos clientes, lo cual demuestra que no es una solución escalable.

El programa “FastPrinter” no incluye una interfaz gráfica para poder administrar o configurar el servicio de impresión, es simplemente un controlador PHP que recibe los tickets a imprimir, por lo tanto dificulta implementar características como, actualizar el logo de impresión del negocio de los clientes, configurar el servidor de impresión local o poder incluso administrar la cola de impresión para ticketeras en red. Tampoco incluye un sistema de logging o registros de depuración, Un sistema de logging se utiliza para diversos fines como depuración, monitoreo y cumplimientos de seguridad, ya que trae información de todo lo que pasa internamente en la ejecución de un programa [6], agregar este sistema a FastPrinter se vuelve complicado por la forma en la que está desarrollado lo cual dificulta el proceso de depuración en caso de errores.

Visto los problemas mencionados anteriormente y que no se encontraron trabajos relacionados se desarrolló una aplicación de escritorio multiplataforma como servicio de impresión de tickets, que implemente un mecanismo de impresión de documentos genéricos, administre la conexión con los diferentes tipos de ticketeras térmicas e implemente una API que pueda ser integrado a los sistemas de punto de venta. El objetivo de este artículo es documentar el proceso de desarrollo de esta aplicación, para que sirva como referencia para el desarrollo de futuros estudios similares, así como evidenciar las mejoras con respecto a FastPrinter y como esto incluso se puede integrar no solo a los sistemas de punto de venta de la empresa PUYU, si no también a cualquier otra solución de software POS. El producto final mejoró todo aspecto técnico de la solución de software de impresión de tickets “FastPrinter”, constituyendo una solución más robusta, con mejores mecanismos de distribución y despliegue a los usuarios finales, actualizaciones automáticas, integración de un mecanismo de logging, interfaces gráficas para la configuración e interacción con el servidor de impresión de tickets y la construcción de un API para la integración con los sistemas de punto de venta.

2 Metodología para la implementación de la aplicación

Es una investigación de tipo aplicada y el alcance es descriptivo. La aplicación desarrollada se integró por completo en el sistema de punto de venta YuRes de la empresa Puyú. Para el diseño y la programación se definió la arquitectura lógica y se utilizó la metodología ágil de desarrollo de software Scrum.

2.1 Arquitectura lógica

La arquitectura lógica representa la interconexión entre las partes que componen la solución de software [7], [8]. La aplicación, a la cual se hará referencia de ahora en adelante como “PukaHTTP”, funciona en una red de área local (LAN), los sistemas de punto de venta consumen la API de la aplicación utilizando el protocolo HTTP y enviando los documentos de impresión (tickets) en formato JSON. Internamente, los documentos son procesados y convertidos a cadenas de bytes según el estándar ESC/POS de Epson, los bytes representan comandos de impresión que las ticketeras térmicas interpretan para emitir los documentos en papel térmico. Los bytes son redirigidos a la ticketera correspondiente a través de cable de red, protocolo Samba, USB o puerto serial. PukaHTTP implementa otros endpoints para la reimpresión y liberación de tickets, así como un controlador WebSocket para el seguimiento en tiempo real del número de elementos en la cola de impresión. Ver Fig. 1.

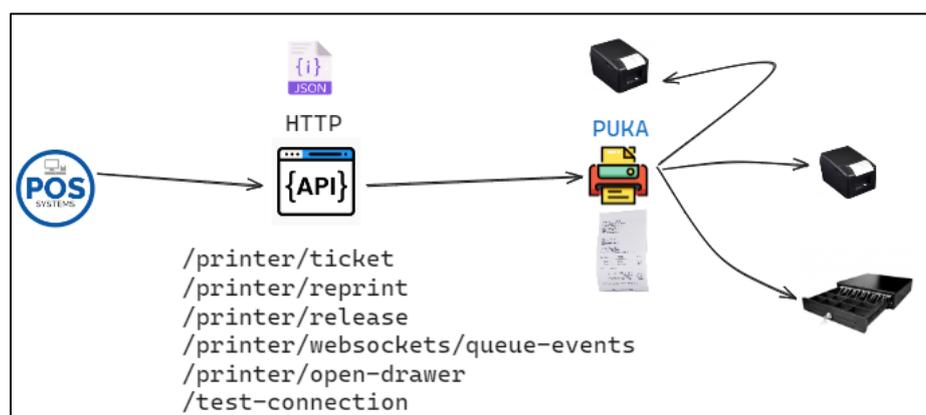


Fig. 1. Arquitectura lógica de la aplicación

2.2 Metodología Scrum

Para el proceso de implementación, se utilizó un enfoque basado en iteraciones o sprints, tal como lo propone la metodología ágil Scrum [9], [10].

Primero se definió el Product Backlog (Tabla 1.) según las funcionalidades que requería la aplicación y en función de ello se inició con el desarrollo de los sprints, cada sprint tuvo una duración de 1 semana, en la que todos los sábados se llevaba a cabo el Sprint Review, donde se definían las tareas de la siguiente semana (Sprint Backlog). Los Daily Sprint fueron de lunes a viernes de 8:30 a.m. a 9:00 a.m. En cada sprint se entregaban elementos funcionales de la aplicación para que fueran evaluados y se obtuviera una retrospectiva de lo que se desarrollaba, y cambiar o actualizar el enfoque si fuese necesario.

Tabla 1. Product Backlog de PukaHTTP

Nº	Item
1	Implementar un mecanismo de conexión con los distintos tipos de ticketeras, mediante Ethernet, protocolo samba, USB, puerto serial, etc.
2	Implementar un servidor http para definir el API de impresión de tickets usando un framework para agilizar el desarrollo de los endpoint y el control de excepciones.
3	Integrar un mecanismo de logging con capacidad de guardar información de la fecha y hora de los posibles incidentes, así mismo con una política de limpieza periódica.
4	Agregar interfaces gráficas para la configuración del servidor, actualizar el logo de impresión y otras configuraciones.
5	Implementar ventana de acciones de impresión (reimprimir, liberar y visualizar el número de elementos en cola de impresión), así como también un módulo de pruebas de compatibilidad e impresión.
6	Integrar mecanismo de despliegue a los usuarios finales mediante el uso de un instalador multiplataforma y de forma automatizada.
7	Agregar un mecanismo de actualizaciones automáticas.

2.3 Interfaces de la aplicación

La ventana principal es la de configuración inicial del servidor, ahí se ingresa el IP y puerto del servidor local http, opcionalmente se puede configurar el logo de impresión del negocio del cliente de punto de venta, ver Fig. 2.



Fig. 2. Ventana inicial PukaHTTP

Adicionalmente, se tiene una ventana para poder reconfigurar el servidor local http y también poder acceder a los archivos de los logs y configuraciones internas de la aplicación tal como se muestra en la Fig. 3.

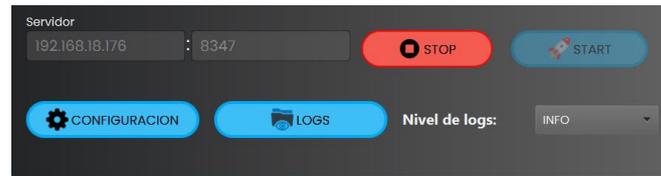


Fig. 3. Ventana de configuración PukaHTTP

Se implementó una interfaz de pruebas de impresión, para descartar incompatibilidades con ciertas ticketeras, como soporte de imágenes, soporte de texto con fondo invertido, soporte de caracteres extraños, conexión exitosa a la ticketera, etc. Así mismo evaluar ciertos aspectos de diseño de los tickets, ver Fig. 4.

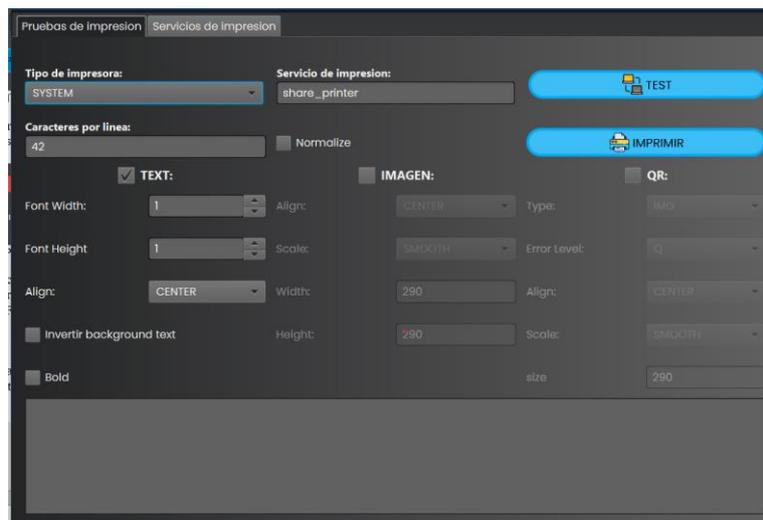


Fig. 4. Ventana de pruebas de impresión PukaHTTP

Finalmente, la Fig. 5 muestra la interfaz principal, que incluye información del número de elementos en cola, acciones de reimpresión o liberación de los elementos en cola de impresión y un botón para acceder a las pruebas de impresión.

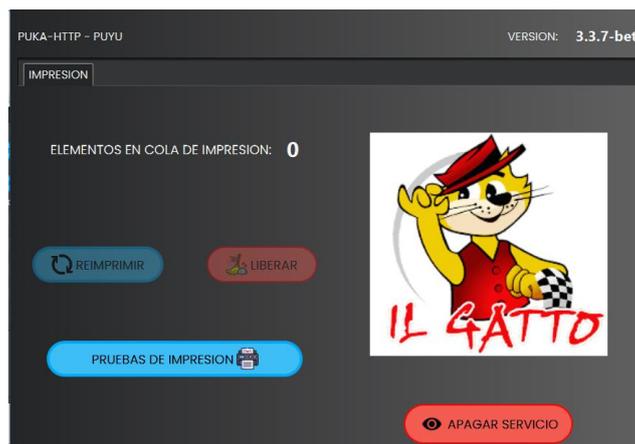


Fig. 5. Ventana principal PukaHTTP

2.4 Despliegue continuo

El despliegue continuo hace referencia a la automatización de llevar a producción el software [11]. Las herramientas utilizadas para este propósito fueron JDeploy y GitHub Actions (G.H.A). JDeploy empaqueta el código en un ejecutable JAR, luego lo envía a un repositorio npm y genera una página de descarga de aplicación en diferentes ejecutables según la plataforma: Linux, Windows o Mac [12]. GitHub Actions es una plataforma de integración y despliegue continuo [13], mediante ella, representamos un flujo de compilación de la aplicación, (ver Fig. 6). Las actualizaciones automáticas ya vienen configuradas gracias a la herramienta JDeploy, cada vez que inicie la aplicación buscará si existe una nueva versión, si lo encuentra, descarga los nuevos paquetes e inicia la aplicación, de lo contrario solo se iniciará la aplicación sin ningún problema.

```
1 name: jDeploy CI with Maven
2
3 on:
4   workflow_dispatch:
5
6 concurrency:
7   group: ${{ github.workflow }}-${{ github.ref }}
8   cancel-in-progress: true
9
10 jobs:
11   build:
12     runs-on: ubuntu-20.04
13     steps:
14       - uses: actions/checkout@v4
15         with:
16           fetch-depth: 0
17       - name: Set package.json.production to package.json
18         run: mv package.production.json package.json
19       - name: Add PRODUCTION file
20         run: touch PRODUCTION
21       - name: Set up JDK 17
22         uses: actions/setup-java@v3
23         with:
24           java-version: '17'
25           distribution: 'temurin'
26           cache: maven
27       - name: Build with Maven
28         run: mvn -B package --file pom.xml
29       - name: Publish App Installer Bundles GITHUB
30         uses: shannah/jdeploy@master
31         with:
32           github_token: ${{ secrets.GH_TOKEN }}
33
34     - run: echo "Final status ${{ job.status }}"
```

Fig. 6. Workflow G.H.A. de despliegue de PukaHTTP

Las pruebas de la aplicación se realizaron en la pizzería Il Gatto en la provincia de Andahuaylas (Fig. 7).



Fig. 7. Pruebas en la pizzería Il Gatto Andahuaylas

3 Resultados

Los resultados de la implementación de la aplicación muestran que los sistemas de punto de venta pueden integrarse a la aplicación instalando el software mediante un instalador multiplataforma sin requerimientos externos, asimismo los sistemas POS pueden consumir el API que implementa la aplicación para la impresión de distintos tipos de documentos. Las pruebas se realizaron en la pizzería Il Gatto, ubicada en la provincia de Andahuaylas, y se pudo observar que la aplicación funciona adecuadamente.

Para la evaluación, se aplicó un cuestionario adecuado según las métricas de usabilidad y calidad de software [14], [15]. El cuestionario consta de 8 preguntas y se muestra en la Figura 8. Está compuesto por 2 bloques, “Calidad de software” que tiene 4 preguntas y “Usabilidad” también con 4 preguntas, luego cada pregunta tiene una valoración que va desde “1” hasta un valor muy positivo “5”.

Calidad de software	1	2	3	4	5
1. ¿La aplicación cumple con las funcionalidades necesarias para la integración con los sistemas de punto de venta?					
2. ¿La aplicación se integra sin problemas con los sistemas de punto de venta?					
3. ¿Qué tan fácil es modificar y actualizar la aplicación?					
4. ¿La aplicación ha sufrido pocos o ningún fallo crítico en el último mes?					
Usabilidad	1	2	3	4	5
1. ¿Qué tan sencillo fue integrar la aplicación de impresión con los sistemas de punto de venta?					
2. ¿La documentación proporcionada fue clara y suficiente?					
3. ¿Qué tan efectivo fue el soporte técnico durante la integración a los sistemas de punto de venta?					
4. ¿Qué tan fácil fue comprender el flujo de trabajo de la aplicación?					

Fig. 8. Cuestionario de usabilidad y calidad de software

El cuestionario se aplicó a 4 desarrolladores de software de la empresa Puyu que integraron la aplicación en los sistemas de punto de venta de la empresa. A cada uno de ellos se les explicó el modo de funcionamiento de PukaHTTP y se dio soporte técnico para su integración en los sistemas POS. El cuestionario fue desarrollado con la herramienta “Google Forms” y se aplicó el 13 de octubre de 2024 cuyos resultados obtenidos se muestran en la Tabla 2.

Tabla 2. Resultados obtenidos del cuestionario

Correo	Pregunta							
	1	2	3	4	5	6	7	8
edwin***@gmail.com	4	5	5	3	4	4	5	5
161***@unamba.edu.pe	5	5	5	4	5	5	5	5
cristhian***@gmail.com	5	5	5	4	5	5	5	5
nahuinlla***@gmail.com	5	5	5	4	4	4	5	5

Del cuestionario aplicado se obtuvo un puntaje de 4.6 en una escala del 1 al 5 en calidad de software y 4.6 en la escala de 1 al 5 en usabilidad. Como promedio general se obtuvo un 4.6 lo que indica que las personas encuestadas están satisfechos con la aplicación.

4 Conclusiones y trabajo futuro

Se desarrolló una aplicación de escritorio para la impresión de tickets en una empresa de desarrollo de software. Para esto se instala la aplicación mediante un instalador multiplataforma sin dependencias, luego los sistemas de punto de venta consumen el API que implementa la aplicación para imprimir varios tipos de documentos en distintas ticketeras ubicadas en diferentes partes de un local. Por otro lado, después se aplicó un cuestionario de usabilidad y calidad de software para evaluar la aplicación; los resultados muestran que se tiene un puntaje de 4.6 en la escala de 1 a 5.

Como trabajo futuro, se podría implementar un servidor de impresión en la nube, que utilice el protocolo WebSockets, de tal forma que se establezca una comunicación segura SSL con el servicio de impresión.

5 Biografías

- Oscar Alcides Choquehuallpa Hurtado, egresado en Ingeniería Informática y Sistemas de la Universidad Nacional Micaela Bastidas de Apurímac.
- Jonathan Bustinza Mendoza, egresado en Ingeniería Informática y Sistemas de la Universidad Nacional Micaela Bastidas de Apurímac.
- Mario Aquino Cruz, Docente en la Universidad Nacional Micaela Bastidas de Apurímac Perú, MSc. en Informática, investigador en las áreas de informática educativa, IoT, inteligencia artificial y ciberseguridad.

6 Referencias

- [1] O. Alfaro, “Migración del punto de venta e infraestructura a la nube para mejora de procesos empresa MONT,” Universidad San Ignacio de Loyola, 2017. Accessed: Oct. 05, 2024. [Online]. Available: <https://hdl.handle.net/20.500.14005/3824>
- [2] Sunat, “Régimen MYPE Tributario - RMT - Orientación - Superintendencia Nacional de Aduanas y de Administración Tributaria - Plataforma del Estado Peruano.” Accessed: Oct. 06, 2024. [Online]. Available: <https://www.gob.pe/8334-ticket-o-comprobante-de-pago-valido>
- [3] “Sistemas con facturación electronica | Puyu.pe.” Accessed: Oct. 14, 2024. [Online]. Available: <https://puyu.pe/>
- [4] C. C. Mohan, S. S. Ahmed, N. V. Priya, M. Jahnavi, and T. P. Babu, “E - Health Centre Maintenance System using PHP with MySQL and XAMPP Web Server,” *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 859–865, Nov. 2022, doi: 10.48175/IJARSCT-7577.
- [5] W. Yang, C. Zhang, M. Pan, C. Xu, Y. Zhou, and Z. Huang, “Do Developers Really Know How to Use Git Commands A Large-scale Study Using Stack Overflow,” *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 3, Apr. 2022, doi: 10.1145/3494518.
- [6] B. Chen and Z. M. J. Jiang, “Studying the use of java logging utilities in the wild,” in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, Jun. 2020, pp. 397–408. doi: 10.1145/3377811.3380408.
- [7] I. Sommerville, *Ingeniería del software*, Séptima. Pearson Addison Wesley, 2005.
- [8] R. Pressman, *Ingeniería del Software. Un Enfoque Practico*, Quinta. Mc Graw Hill, 2001.
- [9] M. Trigas Gallego, “Metodología Scrum,” 2012.
- [10] N. X. Cortes, “Metodología Scrum versus Metodología en Cascada en el Desarrollo de Software,” 2013, Accessed: Oct. 06, 2024. [Online]. Available: <https://prcrepository.org:443/xmlui/handle/20.500.12475/2369>
- [11] C. Sánchez Gómez, “Estudio de herramientas de despliegue continuo de aplicaciones, y sus ventajas competitivas en un mundo marcado por la agilidad,” 2019, Accessed: Oct. 07, 2024. [Online]. Available: <https://hdl.handle.net/10016/29802>
- [12] “Developer friendly desktop deployment tool.” Accessed: Oct. 07, 2024. [Online]. Available: <https://www.jdeploy.com/>
- [13] “Documentación de GitHub Actions - Documentación de GitHub.” Accessed: Oct. 07, 2024. [Online]. Available: <https://docs.github.com/es/actions/about-github-actions/understanding-github-actions>
- [14] N. Bevan, J. Carter, J. Earthy, T. Geis, and S. Harker, “New ISO standards for usability, usability reports and usability measures,” Springer Verlag, 2016, pp. 268–278. doi: 10.1007/978-3-319-39510-4_25.
- [15] M. Lang, *Usability engineering*, vol. 44, no. 1. Academic Press, 2002. doi: 10.1524/itit.2002.44.1.003.