



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

DESARROLLO DE UN SISTEMA WEB PARA LA AUTOMATIZACIÓN DE PEDIDOS EN TIEMPO REAL BAJO EL MODELO DE SOFTWARE COMO SERVICIO (SAAS) PARA LA PIZZERÍA D'BAGGIO

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO DE SOFTWARE

AUTORES: ALEX PATRICIO CUENCA RAMOS

FRANCISCO ALEJANDRO LLININ FLORES

DIRECTOR: Ing. MIGUEL ANGEL DUQUE VIQUE

Riobamba – Ecuador

2022

© 2022, Alex Patricio Cuenca Ramos y Francisco Alejandro Llinin Flores

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Nosotros, Alex Patricio Cuenca Ramos y Francisco Alejandro Llinin Flores, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 25 de noviembre de 2022



Alex Patricio Cuenca Ramos
171827655-1



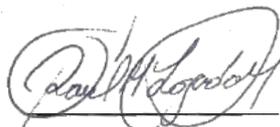
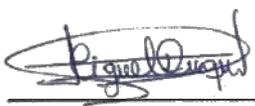
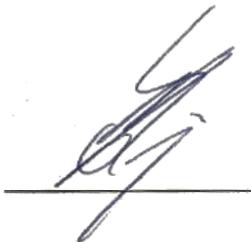
Francisco Alejandro Llinin Flores
060474135-5

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Proyecto Técnico, **DESARROLLO DE UN SISTEMA WEB PARA LA AUTOMATIZACIÓN DE PEDIDOS EN TIEMPO REAL BAJO EL MODELO DE SOFTWARE COMO SERVICIO (SAAS) PARA LA PIZZERÍA D'BAGGIO**, realizado por los señores: **ALEX PATRICIO CUENCA RAMOS Y FRANCISCO ALEJANDRO LLININ FLORES**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Lcd. Raúl Marcelo Lozada Yánez PRESIDENTE DEL TRIBUNAL		2022-11-25
Ing. Miguel Ángel Duque Vique DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2022-11-25
Dr. Omar Salvador Gómez Gómez ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2022-11-25

DEDICATORIA

Dedicamos el presente trabajo al sacrificio constante de nuestros padres y familiares que por años nos han sabido brindar su apoyo incondicional y que gracias a su consejo hemos podido seguir el camino correcto. Gracias a ustedes nos hemos convertido en hombres moral y éticamente correctos. A nuestros amigos y personas más cercanas que en los momentos más difíciles nos brindaron su consejo y sabiduría para seguir adelante y nunca abandonar nuestros sueños y metas.

Alex y Francisco

AGRADECIMIENTO

Queremos agradecer a Dios que nos ha brindado la sabiduría y el conocimiento necesario para poder llegar a este punto de nuestras carreras. A nuestros apacibles padres que con su apoyo incondicional hoy nos ven convertirnos en hombres de bien. A la ilustrísima Escuela Superior Politécnica de Chimborazo que por cinco años nos ha preparado no solo como profesionales sino también como buenos hombres y seres humanos capaces de contribuir con el mundo y nuestra sociedad.

Damos nuestro más sincero agradecimiento al ingeniero Miguel Ángel Duque y al doctor Omar Gómez por sus numerosas formas de apoyo y colaboración en este trabajo de investigación curricular.

Alex y Francisco

TABLA DE CONTENIDOS

_Toc126756876

ÍNDICE DE ILUSTRACIONES.....	XI
ÍNDICE DE ANEXOS.....	XIV
RESUMEN	XV
ABSTRACT.....	XVI
INTRODUCCIÓN	1
CAPÍTULO I	
1. DIAGNÓSTICO DEL PROBLEMA	2
1.1. Antecedentes	2
<i>1.1.1. Formulación del problema.....</i>	<i>3</i>
<i>1.1.2. Sistematización del problema.....</i>	<i>3</i>
1.2. Justificación	3
<i>1.2.1. Justificación teórica</i>	<i>4</i>
<i>1.2.2. Justificación aplicativa.....</i>	<i>5</i>
1.3. Objetivos	6
<i>1.3.1. Objetivo General.....</i>	<i>6</i>
<i>1.3.2. Objetivos Específicos</i>	<i>7</i>
CAPÍTULO II	
2. FUNDAMENTOS TEÓRICOS.....	8
2.1. Servicios Alimentarios.....	8
<i>2.1.1. Proceso de Pedidos</i>	<i>8</i>
<i>2.1.2. Sistemas de Automatización de Pedidos.....</i>	<i>9</i>
<i>2.1.3. Beneficios de la Automatización de Pedidos.....</i>	<i>9</i>
2.2. Computación en la nube.....	10
<i>2.2.1. Software como servicio SaaS.....</i>	<i>11</i>
<i>2.2.2. Beneficios de aplicaciones con modelo SaaS.....</i>	<i>12</i>
<i>2.2.3. Desventajas del modelo SaaS.....</i>	<i>13</i>
<i>2.2.4. Herramientas usadas en el modelo SaaS.....</i>	<i>13</i>
<i>2.2.5. Implementación de software tipo SaaS.....</i>	<i>15</i>
2.3. Herramientas de desarrollo	16
<i>2.3.1. Lenguajes de Programación.....</i>	<i>16</i>
<i>2.3.1.1. Python</i>	<i>16</i>
<i>2.3.1.2. JavaScript.....</i>	<i>16</i>
<i>2.3.2. Frameworks Web.....</i>	<i>17</i>
<i>2.3.2.1. ReactJS.....</i>	<i>17</i>

2.3.2.2. Django.....	17
2.3.3. Base de datos	17
2.3.3.1. PostgreSQL	17
2.3.4. Librerías y/o módulos.....	18
2.3.4.1. Tailwind	18
2.3.4.2. Semantic UI.....	18
2.3.5. Entorno de desarrollo.....	18
2.3.5.1. Visual Studio Code.....	18
2.4. Metodología de desarrollo Kanban	18
2.4.1. Roles de la metodología Kanban	20
2.4.2. Implementación.....	20
2.5. Norma ISO/IEC 25010: Modelos de calidad de sistemas y software	21
2.5.1. Características de la ISO/IEC 25010: 2011	21
2.5.2. Eficiencia de desempeño	22
2.6. Trabajos relacionados.....	23
CAPÍTULO III	
3. MARCO METODOLÓGICO	24
3.1. Tipo de estudio.....	24
3.1.1. Métodos y técnicas.....	24
3.2. Gestión del proyecto.....	25
3.2.1. Análisis previo al desarrollo del proyecto	25
3.2.2. Proceso del negocio de la pizzería	25
3.2.3. Estudio de la factibilidad.....	28
3.2.4. Gestión de riesgos.....	28
3.3. Ingeniería del Producto.....	29
3.3.1. Metodología Kanban para el desarrollo del software	29
3.3.2. Backlog del producto (Product backlog).....	29
3.3.3. Requerimientos.....	30
3.3.4. Diseño	31
3.3.4.1. Arquitectura del sistema	31
3.3.4.2. Base de datos.....	32
3.3.4.3. Interfaces de usuario	37
3.3.5. Desarrollo	42
3.3.5.1. Estándar de codificación.....	42
3.3.5.2. Configuración de los entornos de desarrollo.....	45
3.3.6. Pruebas.....	57

3.3.7. Despliegue	58
3.3.7.1. <i>Back-end</i>	58
3.4. Metodología para la determinación de la eficiencia de desempeño	66
3.4.1. Población y muestra	68
3.4.1.1. <i>Población y muestra de la eficiencia de desempeño</i>	68
3.4.2. Planteamiento de la hipótesis	69
CAPÍTULO IV	
4. RESULTADOS	70
4.1. Eficiencia en el desempeño	70
4.1.1. Comportamiento temporal	70
4.1.1.1. <i>Tiempo de respuesta</i>	70
4.1.1.2. <i>Comparación de los tiempos del proceso</i>	71
4.1.2. Utilización de recursos	74
4.1.2.1. <i>Uso de memoria RAM</i>	75
4.1.2.2. <i>Uso del procesador (CPU)</i>	75
CONCLUSIONES	98
RECOMENDACIONES	99
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-2: Muestreo estratificado para las operaciones.....	12
Tabla 2-2: Desventajas del modelo SaaS en el desarrollo de aplicaciones	13
Tabla 3-2: Herramientas en el mercado del Cloud Computing.....	14
Tabla 4-2: Tipos de datos en Python.....	16
Tabla 5-2: Relación entre el modelo MVC y MVT de Django.....	17
Tabla 6-2: Características y subcaracterísticas de la ISO/IEC 25010 2011	21
Tabla 7-2: Subcaracterísticas de eficiencia de desempeño	22
Tabla 1-3: Métodos, Técnicas y Fuentes para el Trabajo de Integración Curricular.....	24
Tabla 2-3: Diagrama SIPOC del proceso de toma de pedidos.....	25
Tabla 3-3: Hoja de gestión de riesgo.....	29
Tabla 4-3: Requisitos funcionales	31
Tabla 5-3: Diccionario de datos de la tabla User.....	36
Tabla 6-3: Métodos, Técnicas y Fuentes para el Trabajo de Integración Curricular.....	37
Tabla 7-3: Estándar de codificación para la base de datos	42
Tabla 8-3: Estándar de codificación para el back-end	43
Tabla 9-3: Estándar de codificación para el Front-end.....	44
Tabla 10-3: Estándar de codificación para la base de datos.....	47
Tabla 11-3: Indicadores para medir la eficiencia Formato de prueba de aceptación.	57
Tabla 12-3: Indicadores para medir la eficiencia.....	67
Tabla 13-3: Ficha para el comportamiento en el tiempo	67
Tabla 14-3: Ficha para la utilización de recursos	67
Tabla 15-3: Ficha para la utilización de recursos	68
Tabla 16-3: Muestreo estratificado para las operaciones.....	69
Tabla 1-4: Tiempo manual del proceso de pedidos	70
Tabla 2-4: Tiempo de respuesta con My iCard	71
Tabla 3-4: Comparativa de resultados entre el proceso manual y automatizado	72
Tabla 4-4: Cálculo de la diferencia entre los tiempos.....	73
Tabla 5-4: Indicadores de evaluación uso de memoria RAM.....	75
Tabla 6-4: Indicadores de evaluación uso de CPU.....	75
Tabla 7-4: Resultados de las mediciones en el servidor	81
Tabla 8-4: Utilización de memoria y CPU cuando se agrega una mesa	86
Tabla 9-4: Utilización de memoria y CPU cuando se realiza un pedido	87
Tabla 10-4: Utilización de memoria y CPU cuando se supervisa estados del pedido desde el tablero de mandos	88

Tabla 11-4: Utilización de memoria y CPU al supervisar estados del pedido desde la aplicación del cliente.....	89
Tabla 12-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido.....	90
Tabla 13-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido.....	91
Tabla 14-4: Utilización de memoria y CPU al extraer el reporte historial de pagos diarios	92
Tabla 7-4: Resultados de las mediciones de RAM en distintos navegadores	93
Tabla 7-4: Resultados de las mediciones de RAM en distintos navegadores	95
Tabla 7-4: Ponderaciones de las subcaracterísticas a nivel de servidor y cliente.....	96
Tabla 7-4: Resultados de la eficiencia de desempeño	97

ÍNDICE DE ILUSTRACIONES

Ilustración 1-2: Características del Cloud Computing	10
Ilustración 2-2: Modelo de servicios cloud.....	11
Ilustración 3-2: Arquitectura multitenant.....	12
Ilustración 4-2: Adopción de plataformas Cloud en empresas.....	14
Ilustración 5-2: Proceso de implementación de aplicaciones SaaS	15
Ilustración 6-2: Metodología Kanban.....	19
Ilustración 7-2: Reglas principales de Kanban	19
Ilustración 8-2: Flujo de trabajo principal de Kanban.....	20
Ilustración 1-3: Proceso actual de toma de pedidos en la pizzería	26
Ilustración 2-3: Modelo actual del proceso de toma de pedidos actual en BPMN	26
Ilustración 3-3: Proceso de toma de pedidos deseado en la pizzería.....	27
Ilustración 4-3: Modelo propuesto para mejorar la eficiencia en el proceso de toma de pedidos en BPMN	27
Ilustración 5-3: Tablero Kanban del Backlog de “My iCard”.....	30
Ilustración 6-3: Arquitectura Multi-Tenant y Single-Tenant	32
Ilustración 7-3: Diseño y modelado de datos lógico	34
Ilustración 8-3: Diseño y modelado de datos físico	35
Ilustración 9-3: Diseño del wireframe de la página de inicio.....	40
Ilustración 10-3: Diseño del wireframe del inicio de sesión.	41
Ilustración 11-3: Logos de “My iCard”	41
Ilustración 12-3: Selección de paleta de colores primarios para la aplicación “My iCard”.....	42
Ilustración 13-3: Comando para la creación de un nuevo proyecto en ReactJS	45
Ilustración 14-3: Comando para iniciar servidor local en ReactJS	46
Ilustración 15-3: Comando para añadidura de Semantic Ui React y Sass.....	46
Ilustración 16-3: Comando para la importación de Semantic UI React y Sass	46
Ilustración 17-3: Comando para la creación de un nuevo proyecto en Django	47
Ilustración 18-3: Configuración de conexión de una base de datos PostgreSQL en Django.....	48
Ilustración 19-3: Comando para crear un superusuario de un proyecto en Django.....	48
Ilustración 20-3: Comando para crear una aplicación en un proyecto en Django	49
Ilustración 21-3: Configuración de aplicaciones en un proyecto en Django	49
Ilustración 22-3: Clase Payment dentro de payments/models.py.....	50
Ilustración 23-3: Campo “amount” de payments/models.py con una propiedad de validación.....	51
Ilustración 24-3: Sobrecarga del método <code>__str__</code> y definición de la clase Meta	51
Ilustración 25-3: Aplicación Payment dentro del panel de administración de Django.....	51

Ilustración 26-3: Configuraciones del panel de administración de payments/admin.py	52
Ilustración 27-3: Comando para ejecutar migraciones.....	52
Ilustración 28-3: Comando para ejecutar migraciones.....	52
Ilustración 29-3: Fichero payments/routers.py	53
Ilustración 30-3: Serializador basado en el modelo de la clase Payment.....	53
Ilustración 31-3: Componente TablePayments.....	55
Ilustración 32-3: Fichero con los métodos de consumo para los pagos	56
Ilustración 33-3: Vista de la lista que muestra los pagos realizados en el establecimiento	57
Ilustración 34-3: Interfaz de línea de comandos para AWS EB.....	59
Ilustración 35-3: Inicializar un repositorio en AWS EB.....	59
Ilustración 36-3: Fichero config.yml dentro de .elasticbeanstalk.....	60
Ilustración 37-3: Inicializar un entorno en AWS EB.....	60
Ilustración 38-3: Estado de un entorno y aplicación en AWS EB.....	61
Ilustración 39-3: Estado de un entorno y aplicación en AWS EB.....	61
Ilustración 40-3: Estado de un entorno y aplicación en AWS EB.....	61
Ilustración 41-3: Consola del entorno sobre el cual se ejecuta “My iCard”	62
Ilustración 42-3: Fichero django.config dentro de “.ebextensions”	62
Ilustración 43-3: Fichero django.config dentro de “.ebextensions”	63
Ilustración 44-3: Configuración de base de datos en AWS RDS.....	64
Ilustración 45-3: Fichero django.config dentro de “.ebextensions”	64
Ilustración 46-3: Fichero “.securelistener-clb.config” dentro de “.ebextensions”	65
Ilustración 47-3: Configuración ALLOWED_HOSTS dentro de settings/prod.py.....	65
Ilustración 48-3: Configuración SSL dentro de settings/prod.py.....	66
Ilustración 49-3: Configuración de base de datos en AWS RDS.....	66
Ilustración 1-4: Comparativa de tiempos por funcionalidades	72
Ilustración 2-4: Porcentaje de los tiempos de respuesta.....	73
Ilustración 3-4: Prueba t de Student.....	74
Ilustración 4-4: Cantidad de peticiones HTTP 2XX cuando se agrega una mesa	76
Ilustración 5-4: Utilización de memoria RAM y CPU cuando se agrega una mesa.....	76
Ilustración 6-4: Cantidad de peticiones HTTP 2XX cuando se realiza un pedido.	77
Ilustración 7-4: Utilización de memoria RAM y CPU cuando se realiza un pedido.	77
Ilustración 8-4: Cantidad de peticiones HTTP 2XX cuando se supervisa estados del pedido... 78	
Ilustración 9-4: Utilización de memoria RAM y CPU cuando se supervisa estados del pedido desde el tablero de mandos.....	78
Ilustración 10-4: Cantidad de peticiones HTTP 2XX cuando se supervisa estados del pedido desde la aplicación del cliente.	78

Ilustración 11-4: Utilización de memoria RAM y CPU cuando se supervisa estados del pedido desde la aplicación del cliente.	79
Ilustración 12-4: Cantidad de peticiones HTTP 2XX cuando se genera la cuenta de un pedido	79
Ilustración 13-4: Utilización de memoria RAM y CPU cuando se genera la cuenta de un pedido	80
Ilustración 14-4: Cantidad de peticiones HTTP 2XX cuando se genera la cuenta de un pedido	80
Ilustración 15-4: Utilización de memoria RAM y CPU cuando se genera la cuenta de un pedido	80
Ilustración 16-4: Cantidad de peticiones HTTP 2XX cuando se extrae el reporte historial de pagos diarios.	81
Ilustración 17-4: Utilización de memoria RAM y CPU cuando se extrae el reporte historial de pagos diarios.	81
Ilustración 18-4: Utilización de RAM en el servidor de AWS por cada funcionalidad	82
Ilustración 19-4: Utilización del CPU en el servidor de AWS por cada funcionalidad.....	82
Ilustración 20-4: PID de la ventana del navegador Microsoft Edge donde se ejecuta el sistema administrador	84
Ilustración 21-4: PID de la ventana del navegador Opera Internet donde se ejecuta el sistema administrador	85
Ilustración 22-4: PID de la ventana del navegador Mozilla Firefox donde se ejecuta el sistema administrador	86
Ilustración 23-4: Utilización de memoria y CPU cuando se agrega una mesa.....	87
Ilustración 24-4: Utilización de memoria y CPU cuando se realiza un pedido.	88
Ilustración 25-4: Utilización de memoria y CPU cuando se supervisa estados del pedido desde el tablero de mandos.....	89
Ilustración 26-4: Utilización de memoria y CPU cuando se supervisa estados del pedido desde la aplicación del cliente.	90
Ilustración 27-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido.	91
Ilustración 28-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido.	92
Ilustración 29-4: Utilización de memoria y CPU cuando se extrae el reporte historial de pagos diarios.	93
Ilustración 30-4: Comparativa del uso de RAM en diferentes navegadores	94
Ilustración 31-4: Comparativa del promedio de uso de RAM	94
Ilustración 32-4: Comparativa del uso del procesador en diferentes navegadores	95
Ilustración 33-4: Comparativa del promedio de uso del procesador	96
Ilustración 34-4: Resultado de la eficiencia de desempeño	97

ÍNDICE DE ANEXOS

ANEXO A: Documento validación de entrega del Manual técnico y de usuario

ANEXO B: Evolución mensual del Tablero Kanban

ANEXO C: Documento validación del sistema “My iCard”

RESUMEN

El objetivo que se planteó en el presente trabajo de integración curricular fue desarrollar un sistema web para la automatización de pedidos en tiempo real bajo el modelo de software como servicio para la pizzería D' Baggio. Se emplearon los métodos analítico, deductivo y estadístico, la revisión de documentación mediante la web, el diagrama de Modelo y Notación de Procesos de Negocio (BPMN) y la prueba t de Student fueron las técnicas de recolección de datos utilizadas. Entre las herramientas de software aplicadas destacan el editor de código Visual Studio Code, lenguaje de programación Python junto con la tecnología de Django para la construcción del back end, también se manejó JavaScript con la tecnología ReactJs para la construcción del front end; para la base de datos se usó PostgreSQL y para el despliegue se ocupó AWS Elastic Beanstalk. Se empleó la metodología Kanban, para mantener una organización adecuada en la implementación del software. Además, se aplicó el estándar ISO/IEC 25010 para medir la eficiencia de desempeño del sistema. Los resultados obtenidos, en cuanto a la eficiencia de desempeño es del 90%. Mediante la aplicación de la prueba t de Student con un nivel de significancia del 5% y cuatro grados de libertad, determinaron que el tiempo de respuesta del sistema menora en aproximadamente 37.16 minutos al tiempo promedio del proceso manual del negocio, lo que significa un 84.45% de ahorro en el proceso de ordenes en la pizzería. Estos resultados permiten concluir que el sistema cumple con los diferentes requerimientos del cliente y se recomienda promover el uso de los estándares para garantizar la calidad del proceso y producto de software, y también utilizar el navegador Opera Internet para una mejor eficiencia en el manejo del sistema desarrollado.

Palabras clave: <SISTEMA WEB>, <SERVICIOS ALIMENTARIOS>, <METODOLOGÍA KANBAN>, <SOFTWARE COMO SERVICIO (SAAS)>, <ISO/IEC 25010>, <EFICIENCIA>.



ABSTRACT

The objective of this curricular integration work was to develop a web system for the automation of orders in real time under the software as a service model for the D' Baggio pizzeria. We utilized analytical, deductive and statistical methods, web-based documentation review, Business Process Model and Notation (BPMN) diagram and Student's t-test were the data collection techniques utilized. The software tools utilized included Visual Studio Code code editor, Python programming language together with Django technology for building the back end, JavaScript with ReactJs technology was also utilized for building the front end; we utilized PostgreSQL for the database and AWS Elastic Beanstalk for deployment. We utilized the Kanban methodology as well to maintain an adequate organization in the software implementation. Furthermore, we applied the ISO/IEC 25010 standard to measure the system's performance efficiency. The results obtained, in terms of performance efficiency is 90%. By applying Student's t-test with a significance level of 5% and four degrees of freedom, we determined that the response time of the system decreases by approximately 37.16 minutes the average time of the manual process of the business, which means 84.45% savings in the order process in the pizzeria. These results allow us to conclude that the system meets the different requirements of the client and it is recommended to promote the use of standards to guarantee the quality of the process and software product, and also to utilize the Opera Internet browser for a better efficiency in the management of the developed system.

Key words: <WEB SYSTEM>, < FOOD SERVICES >, <KANBAN METHODOLOGY>, <SOFTWARE AS SERVICE (SAAS)>, <ISO/IEC 25010 STANDARD>, <EFFICIENCY>.

Translated by:



Lic. Carolina Campaña D. Mgs.

ID number: 1804191482

EFL Professor

INTRODUCCIÓN

En el tiempo actual los sistemas web han sido introducidos en una gran cantidad de áreas sociales permitiendo la automatización de diferentes procesos que se realiza de manera manual, donde se ve que ha dado resultados como la optimización de recursos, tiempo y costos. En el área de los servicios alimentarios tras la época postpandemia de la Covid-19 se ha tenido varias innovaciones que facilitan a los consumidores realizar pedidos respetando las normas de bioseguridad.

La mayoría de pequeñas y medianas empresas del servicio alimentario en el territorio ecuatoriano aún realizan el proceso de la toma de pedidos de manera manual, provocando una demora en la atención debido a factores como la pérdida de órdenes y la demora en generar una cuenta del pedido por los cálculos que los llevan manualmente. De esta forma se ha visto la necesidad de la implementación de un sistema web para la toma de pedidos denominado “My iCard”, el mismo que al ser una solución cloud servirá como apoyo a varias empresas que lo necesiten ayudando mejorar la atención al cliente y por ende impulsar este sector en la economía del país.

Con respecto a la estructuración de este documento se encuentra dividido en cuatro capítulos que se los detalla a continuación:

CAPÍTULO I: Menciona los antecedentes de la problematización con su respectiva justificación teórica y aplicativa, también se menciona los objetivos generales y específicos

CAPÍTULO II: Presenta la conceptualización teórica de los temas de servicios alimentarios, características del servicio, Cloud Computing, Metodología Kanban, herramientas de desarrollo y la Norma ISO/IEC 25010.

CAPÍTULO III: Detalla el tipo de estudio, métodos y técnicas usadas para la evaluación de los objetivos propuestos, se describe el tipo de muestra e instrumentos aplicados para medir el nivel de eficiencia y la aplicación de la metodología en el desarrollo del producto software.

CAPÍTULO IV: Presenta los resultados obtenidos en relación con los objetivos específicos propuestos. Finalmente, se expone las conclusiones y recomendaciones del proyecto realizado.

CAPITULO I

1. DIAGNÓSTICO DEL PROBLEMA

Esta sección aborda los antecedentes encontrados sobre el tema a desarrollar, brindando una justificación teórica y práctica de este, proponiendo varios objetivos que se alcanzaran una vez concluido el Trabajo de Integración Curricular.

1.1. Antecedentes

La pizzería D' Baggio es una empresa dedicada a brindar servicios alimentarios como platos italianos y pizzas, creada el 25 de enero 1999 pero cerrada el 20 de diciembre del mismo año, tras el retorno del Sr. Jorge Washington Flores Córdor, la pizzería reinicia las actividades desde el 14 de Julio del 2005 hasta la actualidad. Según la página (Tripadvisor, 2021) la empresa ocupa el puesto número once a nivel de restaurantes en Riobamba, cuenta con alrededor de diez trabajadores en diferentes áreas, debido a la época de postpandemia el servicio necesita una adaptación a las nuevas demandas sociales.

Bajo este contexto se desea desarrollar un sistema web para la automatización de los pedidos en tiempo real bajo el modelo SaaS que según (Bravo 2009) es un modelo de distribución del software que proporciona a los clientes el acceso a aplicaciones a través de la Internet, cuyo fin será la adaptación del servicio prestado tradicionalmente a algo más eficiente y acorde al contexto actual, donde mediante el uso de códigos QR y el propio celular de los clientes se pueda realizar los pedidos de manera rápida y segura, los mismos que llegarán al administrador en tiempo real al control de mandos de su equipo. Además, esto permitirá contar con un respaldo de los pedidos evitando así la gestión lenta y poco eficiente en los pedidos de la pizzería.

Según (Romero et al. 2021) menciona que a partir del año 2020 tras la pandemia por Covid-19 el mundo tomó una nueva realidad, y más aún en los servicios alimentarios. Se observa que los locales que brindan este servicio deben contar con medidas de bioseguridad tanto en la toma de órdenes como en el pago, provocando dificultades como el tiempo excesivo que se toma un mesero en atender una orden de varias personas, la pérdida de comandas que no llegan a la cocina ni a la caja, además el establecimiento no cuenta con un sistema informático que gestione este proceso, todo esto ha desencadenado en la mala eficiencia en el proceso de pedidos al seguir realizando esto de manera tradicional. De continuar este problema seguirá

provocando disconformidad en los clientes por la demora de la atención, así también se verá reflejado un bajo crecimiento económico y pérdidas significativas pues la empresa no cuenta con una herramienta tecnológica acorde a los desafíos actuales.

Existen varios trabajos que se enfocan en esta temática en los que cabe mencionar a (Mendoza 2019) con el tema “Desarrollo de un sistema web, que permita administrar pedidos, aplicada a la Distribuidora Repuestos Córdoba.” Así también el de (Chicaiza 2020) con el tema “Desarrollo de un sistema vía Web para automatizar los pedidos y reservaciones de comidas en la Parrilla “Don Carlos” del cantón Milagro” , ambos trabajos buscan la automatización de pedidos tanto en el aspecto comercial como en servicios alimentarios, usando la metodología SCRUM. El presente proyecto trabajará el sistema bajo el modelo SaaS lo que hace un aporte considerable a los temas citados, mientras que la metodología a emplear es Kanban.

Por lo expuesto anteriormente, se considera necesario realizar un sistema web que logre automatizar los pedidos de manera eficiente. El administrador de la pizzería D’ Baggio hará uso del sistema de manera directa pues es quien generará los códigos QR acorde a las necesidades de la empresa y tendrá en tiempo real todos los pedidos que se hagan en un determinado intervalo de tiempo, los clientes usarán el sistema de manera indirecta pues al ingresar al local podrán realizar el pedido leyendo con su dispositivo el código QR.

1.1.1. Formulación del problema

¿Cómo un sistema web bajo el modelo SAAS mejorará la eficiencia del proceso de pedidos presenciales en la pizzería D’ BAGGIO?

1.1.2. Sistematización del problema

- ¿Cuál es el proceso para realizar un pedido en la pizzería?
- ¿Qué beneficios trae adoptar sistemas bajo el modelo de software como servicio SaaS?
- ¿Qué herramientas son necesarias para adaptar el modelo de software como servicio SaaS?
- ¿Cómo implementar un sistema bajo el modelo de software como servicio SaaS?
- ¿Cómo se va a medir la eficiencia del sistema web para la automatización de pedidos?

1.2. Justificación

1.2.1. Justificación teórica

Para el desarrollo de un sistema web es necesario la utilización de varias herramientas, obteniendo como resultado final un sistema funcional que facilite los procesos dentro de cualquier negocio. Es así como, dentro de este punto se explicará todos los recursos empleados que sustentan el desarrollo del sistema propuesto.

Según (Hernanz & Denham, 2013) un software de automatización de pedidos para restaurantes debe abarcar numerosas alternativas de control, además de ser flexible puesto que las necesidades de un restaurante de clase A no son las mismas que las de un servicio de comida rápida. En este sentido este tipo de sistemas deben ser una herramienta moldeable a las necesidades de control y gestión que precisa el cliente. Según (Llerena y Guillen 2020) como primer paso de la metodología Kanban se debe definir la cantidad ideal de productos a entregar. Esta metodología tiene como objetivo primordial determinar tareas por hacer y cambiar su prioridad en función de los acontecimientos, se busca entonces que la cadena de trabajo sea visible para todos (Banco Bilbao Vizcaya Argentaria, S.A. 2019).

SaaS entrega software y datos como un servicio sobre internet que corre del lado del cliente sin tener que instalarlo en cualquier dispositivo. Esto permite que una sola copia del software se esté ejecutando en un ambiente controlado por los desarrolladores. Esta característica es fundamental dado que se puede mejorar el software y el hardware siempre y cuando no se vulnere la interface de programas de aplicación de lado del cliente (Chávez et al. 2012). Según (Maldonado 2009) SaaS también ofrece a las empresas que lo usan poder liberar recursos, centrarse en su actividad, simplificar la gestión, reducir el tiempo de llegada y ganar productividad facilitando el acceso a la aplicación en cualquier momento y lugar.

Actualmente tenemos a disposición varias tecnologías para el desarrollo como JavaScript que es un lenguaje de programación que se utiliza para la creación de páginas web dinámicas (Eguiluz 2009) , también se destaca ReactJS que es un framework que se utiliza para crear interfaces de usuario dirigidas al frontend (Rawat y Mahajan 2020). De igual forma Django que es un framework para el desarrollo web usando Python (Liawatimena et al. 2018).

A continuación, se describe brevemente las tecnologías que se utilizarán para la realización de este sistema web. Se hará uso de un sistema gestor de base de datos relacional y los lenguajes de programación Python y Javascript para el back-end de la aplicación mediante el empleo del framework Django. En la parte del front-end se hará uso del framework ReactJS y varios complementos más.

1.2.2. Justificación aplicativa

La implementación de un sistema web para la automatización del proceso de pedidos puede contribuir a simplificar dicha actividad dentro de la empresa “D’Baggio”, mejorando la eficiencia de este proceso. De esta manera la entidad tendrá un conocimiento concreto de la ordenes realizadas por los clientes como resultado del uso del sistema, además podrá evitar perjuicio económico generado ante la pérdida de comandas físicas. También la gerencia del establecimiento dispondrá de la información necesaria para la toma de decisiones para ver si tiene productos disponibles que pueda ofrecer a los clientes.

Se logrará desarrollar un sistema web para la automatización de pedidos en la pizzería “D’Baggio” de manera eficiente, puesto que con la ayuda de la metodología KANBAN se cumplirá los objetivos de forma adecuada y rápida, para que exista un cambio medible en este proceso de la empresa. Además, con el enfoque del modelo SaaS, el Administrador del local podrá usar el sistema en su establecimiento de manera efectiva, adicionalmente se tomará en cuenta todos los beneficios que prestará este enfoque para la consecución de los objetivos.

El sistema web constara de lo siguiente módulos que se presentarán a continuación:

- **Módulo de Autenticación:** En el módulo de autenticación los usuarios serán capaces de acceder a las siguientes operaciones: registrar una nueva cuenta activar dicha cuenta mediante el uso de correo electrónico validar y verificar correctamente su cuenta iniciar sesión mediante las credenciales previstas en el registro, así como también recuperar la contraseña mediante el uso de su correo electrónico.
- **Módulo de Categorías:** El módulo categorías pretende abstraer la clasificación de alimentos, bebidas, etc. Que se presentan en el restaurante al cual está dirigido a la aplicación. El módulo será encargado de las operaciones tipo CRUD que se llevarán a cabo sobre el modelo categoría. Estas operaciones incluyen añadir, eliminar, actualizar, actualizar parcialmente, y consultar. Este módulo además estará relacionado con el módulo de productos.
- **Módulo de Pedidos:** El módulo pedido pretende abstraer la operación de solicitar un alimento dentro del restaurante. Será encargado de gestionar las operaciones tipo CRUD que se llevaran a cabo sobre el modelo pedido. Estas operaciones incluyen añadir, eliminar, actualizar, actualizar parcialmente y consultar. Este módulo además estará relacionado con el módulo de mesas y productos.

- **Módulo de Mesas:** El módulo de mesas pretende abstraer la cantidad de lugares disponibles que se encuentran dentro del restaurante. El módulo correspondiente a las mesas será el encargado de gestionar las operaciones tipo CRUD que se llevará cabo sobre el modelo de mesas estas operaciones incluyen añadir, eliminar, actualizar y dar por servida una mesa.
- **Módulo de Productos:** El módulo de productos pretende abstraer los alimentos que se expenden dentro del restaurante. El módulo correspondiente a los productos será el encargado de gestionar las operaciones tipo CRUD que incluyen añadir, eliminar, actualizar, actualizar parcialmente, añadir la cantidad disponible y consultar sobre los sobre el modelo de productos. Este módulo además está relacionado con el módulo de pedidos.
- **Módulo de Usuarios:** El módulo de usuarios pretende abstraer todos aquello todo aquello considerado como un recurso humano es decir clientes, trabajadores, administradores. Este módulo estará encargado de gestionar las operaciones tipo CRUD que incluyen añadir, eliminar, actualizar, actualizar parcialmente y consultar sobre el modelo de usuarios.
- **Módulo de Historial de Pagos:** El módulo del historial de pagos pretende abstraer la forma en la en la que se paga el servicio ofrecido dentro del restaurante. Este módulo estará encargado de gestionar las operaciones tipo Cruz que incluyen añadir, actualizar, eliminar y verificar que un pago haya sido realizado correctamente. Este módulo estará relacionado directamente con el modelo de mesas y pedidos.

El desarrollo del presente Trabajo de Integración Curricular está enfocado en la línea de investigación de la Escuela Superior Politécnica de Chimborazo correspondiente a Tecnologías de la información y comunicación, bajo el programa de Ingeniería de software, en el ámbito de Análisis y diseño de software. Con respecto al plan de nacional de desarrollo de la temática propuesta se enmarca en el eje Social, con el objetivo (7) que busca potenciar las capacidades de la ciudadanía y promover a la educación innovadora, inclusiva y de calidad en todos los niveles. Además, está relacionada con la política (7.4) que busca fortalecer el Sistema de Educación Superior bajo los principios de libertad, autonomía responsable, igualdad de oportunidades, calidad y pertinencia; promoviendo la investigación de alto impacto.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema web para la automatización de pedidos en tiempo real bajo el modelo de software como servicio (SAAS) para la pizzería D'BAGGIO.

1.3.2. Objetivos Específicos

- Establecer el proceso que lleva actualmente el establecimiento para manejar los pedidos en época post pandemia.
- Investigar sobre características, beneficios, desventajas y proceso de implementación que conlleva aplicar el modelo SaaS en el desarrollo de software.
- Desarrollar los módulos de Autenticación, Categorías, Mesas, Productos, Usuarios, Pedidos e Historial de Pagos.
- Evaluar la eficiencia del sistema web mediante la aplicación del estándar ISO/IEC 25010.

CAPITULO II

2. FUNDAMENTOS TEÓRICOS

El propósito de esta sección es realizar una revisión bibliográfica sobre los conceptos generales que intervienen en el Trabajo de Integración Curricular permitiendo que el lector tenga una idea clara sobre la temática, incluyendo las herramientas, metodologías y estándares utilizados en el desarrollo del sistema web.

2.1. Servicios Alimentarios

Acorde con (Tapia et al., 2019) hoy en día las necesidades y el nivel de exigencia de las personas se han elevado de manera exponencial, debido a factores como la globalización, por tanto, la calidad del servicio debe evolucionar acorde a estos requerimientos; generando una mejora en la prestación óptima del servicio para las personas. Además, menciona que el servicio de alimentación y bebidas se ramifica en algunas clasificaciones, tales como: el servicio de restaurantes, eventos sociales, empresas de catering, entre otros.

En el Ecuador el servicio de alimentos y bebidas tiene una connotación importante, ya que según datos del (INEC, 2014), el 24 % de las personas ocupadas en el mismo lo hacen en hoteles y restaurantes. Tras la creación de nuevos establecimientos esta cifra va en aumento, representando una fuerza laboral considerable para el país, por lo tanto, los prestadores de servicios deben poseer los conocimientos necesarios para desenvolverse en este campo.

2.1.1. *Proceso de Pedidos*

De acuerdo con (García, 2018) el pedido tiene un recorrido particular dentro del restaurante. El orden exacto dependerá del tipo y tamaño del negocio, sin embargo, el proceso es básicamente el mismo. Una vez que el encargado ha terminado de tomar la orden, debe entregar inmediatamente una copia al responsable de despachar el pedido. Este deberá a su vez delegar las responsabilidades que le corresponden a cada servicio: las bebidas, la preparación de la comida u otros que el local maneje.

Según (Gallego, 2002) especifica que una comanda es un documento donde el encargado toma el pedido a los clientes de todo aquello que forma parte de la oferta del local. Para ello, previamente se habrá entregado a cada persona que se encuentre en la mesa un menú.

En el proceso de pedidos, la toma de la orden es el primer contacto que el cliente tiene con el establecimiento, por eso es muy importante cuidar todos los detalles para que la experiencia sea positiva para ambas partes, pues según (De Bonis, 2019) esta es una parte crítica en el servicio de alimentos y bebidas, pues, si no sabemos exactamente lo que quiere el cliente, no podremos satisfacerlo.

2.1.2. Sistemas de Automatización de Pedidos

Según (Hernanz & Denham, 2013) un sistema de automatización de pedidos es una herramienta software moldeable a las necesidades de control y gestión que precisa el cliente para automatizar este proceso en su negocio.

Según (IBM, 2021) las características que debe tener un sistema de automatización de pedidos son:

- **Visibilidad:** en esta característica debe existir la posibilidad de ver toda la cadena de suministro y desarrollar procesos más eficientes.
- **Inteligencia:** debe permitir el ajuste de los procesos de gestión de pedidos según las reglas empresariales y los objetivos de rendimiento de la organización.
- **Flexibilidad:** posee la capacidad de segmentar los pedidos en elementos de trabajo únicos que se pueden canalizar a los recursos adecuados.
- **Inventario en tiempo real:** proporciona una vista única del inventario, mostrando lo que hay en stock y los niveles actuales de demanda.
- **Planificación de servicio:** permite que las solicitudes de servicio se traten de forma más eficiente.
- **Tecnologías de participación del cliente:** proporciona al personal que trata con los clientes un punto de vista del cliente, los recursos y el inventario, para que puedan ejecutar transacciones de forma más eficiente.
- **Optimización de la tramitación:** Debe ayudar a la obtención de los datos y recomendar opciones que tengan en cuenta cómo el tiempo de entrega y el costo.

2.1.3. Beneficios de la Automatización de Pedidos

De acuerdo con (Zamora, 2018) toda organización debe estar en evolución paralela con el medio donde se desarrolla, por lo que ésta debe someterse a los cambios que se podrían implicar,

proyectando así un fortalecimiento de todo lo que la institución puede mejorar a nivel de recursos, competencias empresariales y sus alcances.

Existen varios beneficios de automatizar pedidos como los que menciona (Orbit Logistics 2020) :

- **Productividad incrementada:** los datos se liberan más rápidamente debido a la automatización del sistema ahorrando recursos, como mano de obra y papel.
- **Más agilidad:** se debe a la transmisión de información de rutina, como facturas, notificaciones de envío y recepción y, por supuesto, envío de pedidos.
- **Mayor satisfacción del cliente:** recopila toda la información de los clientes, mostrando qué estrategias deben usarse y qué productos son más relevantes para un público específico provocando que el proceso de ventas se vuelve más ágil, aumentando la eficiencia del servicio y la satisfacción del cliente.
- **Mayor ahorro de costos:** permite una mayor productividad reduciendo los costos de la empresa debido a que proporciona un mejor control sobre las actividades, evitando errores.

2.2. Computación en la nube

Según (Palos-Sanchez, Arenas y Mariano, 2017) Cloud es el símbolo que se usa generalmente para representar Internet y Computing reúne los conceptos de informática, lógica de coordinación y almacenamiento. Acorde con (Miralles, 2010) la computación en la nube es una arquitectura de prestación de servicios. Según los analistas, en los próximos años se consolidará tanto entre los usuarios particulares de la red y servicios en línea, como entre las empresas; en ambos casos afectará a su manera de utilizar las Tics en cualquier tipo de negocio.

Hay 5 características del Cloud Computing que se detallan en la **Ilustración 1-2**.



Ilustración 1-2: Características del Cloud Computing

Fuente: (Miralles R., 2010)

Por otra parte, la **Ilustración 2-2** presenta los diferentes modelos de servicio Cloud divididos en tres grandes categorías:

- *IaaS*, una solución que consiste en proveer y gestionar recursos tales como servidores, almacenamiento y redes a través de Internet.
- *PaaS*, un modelo que proporciona un entorno de desarrollo listo para usar, en el que los desarrolladores pueden centrarse en escribir y ejecutar código para crear aplicaciones.
- *SaaS*, consiste en distribuir aplicaciones en la nube a usuarios a través de Internet.

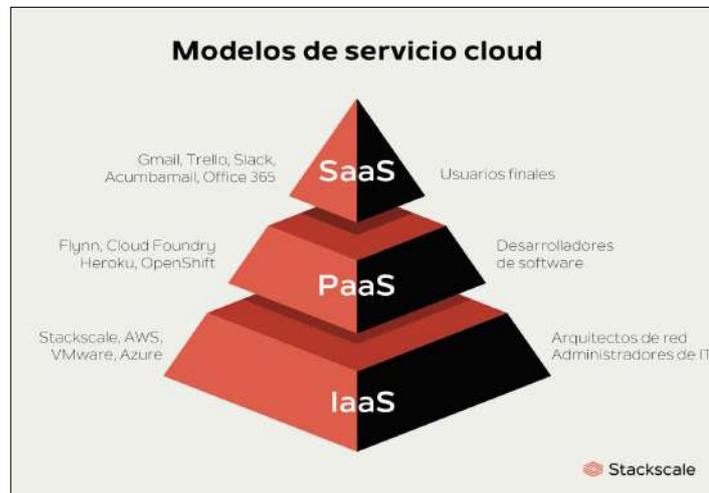


Ilustración 2-2: Modelo de servicios cloud

Fuente: (Stackscale, 2021)

2.2.1. *Software como servicio SaaS*

SaaS es un modelo de distribución del software que proporciona a los clientes el acceso a aplicaciones a través de Internet, de manera que el usuario no tiene que preocuparse del mantenimiento de dichas aplicaciones. Para el usuario, este modelo permite optimizar costes y recursos (Bravo, 2009).

Se considera que este modelo es un método de otorgamiento de licencias y entrega de software a pedido, basado en una solución de alojamiento centralizado que absuelve al cliente de los costos de hardware que conllevan las soluciones alojadas en servidor (Resceanu, Reşceanu y Simionescu, 2014).

La arquitectura en la que se basa el modelo SaaS es conocida como Arquitectura Multi-Tenant como se observa en la **Ilustración 3-2**, la cual consiste en una arquitectura orientada a la nube que implica una sola instancia de la aplicación, pero sirviendo a múltiples clientes u organizaciones. La idea básica de este tipo de arquitecturas es tener una sola base de código que

se ejecuta para todos los clientes, con una misma estructura de datos pero que se diferencie para cada entidad siendo una aplicación común para todos los clientes.

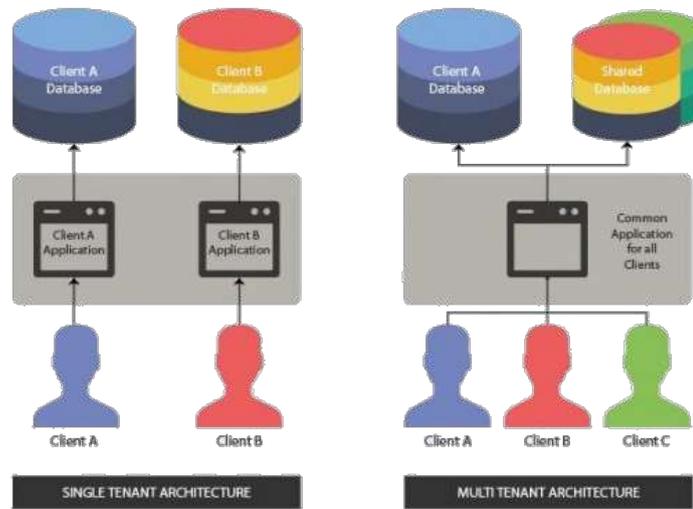


Ilustración 3-2: Arquitectura Multi-Tenant

Fuente: (Alonso A., 2018)

2.2.2. Beneficios de aplicaciones con modelo SaaS

En general, el modelo SaaS requiere menos trabajo técnico a nivel de empresa en comparación con el modelo local tradicional. Al momento de describir beneficios (IONOS, 2019) menciona que el software como servicio se está consolidando como una atractiva solución en la nube ofreciendo varias ventajas frente al software local como las que se detallan en la **Tabla 1-2**.

Tabla 1-2: Muestreo estratificado para las operaciones

Característica	Beneficio
Rapidez	Rápida puesta en funcionamiento y prestación de servicios para clientes.
Mantenimiento	El mantenimiento por parte del proveedor supone un alivio para los trabajadores.
Integración	Sencilla integración de nuevos trabajadores.
Recursos	Utilización de los servicios desde diferentes dispositivos, fijos o móviles.
Arquitectura	La arquitectura en la nube ofrece a todos los usuarios los mismos estándares de seguridad.
Actualizaciones	Nuevas funciones y actualizaciones se implementan e integran mucho más rápido.
Facilidad de uso	Presenta una orientación al usuario provocando que el uso de estas herramientas del modelo sea fácil de entender y usar por cualquier cliente.

Fuente: (Saraswat, M ; Tripathi R., 2020)

Realizado por: Cuenca A.; Llinin F., 2022.

2.2.3. Desventajas del modelo SaaS

La desventaja más clara de SaaS es la dependencia del internet ya que si la empresa está en un área donde los servicios de conectividad son difíciles o si el servicio de internet es malo un software en sitio podría ser mejor para trabajar óptimamente. Además, no siempre se le puede tener completa confianza a un servicio ya que es difícil de imaginar que todos los datos de una empresa se lo almacenen en aplicaciones de terceros (Macías, 2018).

Según (Gomez, 2022) el modelo SaaS presenta algunas desventajas a nivel de funcionalidades de un sistema como los que se detallan en la **Tabla 2-2**.

Tabla 2-2: Desventajas del modelo SaaS en el desarrollo de aplicaciones

Característica	Desventaja
Dependencia	Se debe usar una computadora o dispositivo electrónico con acceso a internet por lo que cuando se acceda al software siempre se cuente con un servicio de conectividad robusto, para que no tener problemas al momento de ejecutar la aplicación.
Escalabilidad	El usuario no tiene permisos ni la posibilidad de hacer modificaciones a las aplicaciones; es decir, es el proveedor del servicio el que tiene la potestad de añadir funcionalidades y modificar los códigos de fuente de estas.
Tolerancia a fallos	Este tipo de software está en ubicaciones centralizadas por lo que si se presenta una falla el servicio se verá interrumpido en todos los usuarios que tienen acceso a esa aplicación puesto que esta aplicación depende de terceros.

Fuente: (Gomez, I., 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

2.2.4. Herramientas usadas en el modelo SaaS

Al implementar una solución cloud se toma en cuenta el lugar en el cual las aplicaciones serán desplegadas. Actualmente existen varias plataformas en la nube que brindan un sinnúmero de herramientas para que sistemas desarrollados se los escale a nivel de nube. Como se puede observar en la **Ilustración 4-2**, se denota una gran competencia entre AWS y Azure, después seguida de Google Cloud. Es importante tener en cuenta el aumento de otras nubes, por ejemplo, Oracle Cloud que tuvo el mayor crecimiento porcentual duplicándose en 2018 y aumentando un 60 % en 2019.

Actualmente existe una gran diversidad de plataformas para trabajar en entornos de Cloud Computing. Según (Dignan, 2021) en lo que va del año 2022, el top 3 de proveedores de servicios lo ostentan Amazon Web Services, Microsoft Azure y Google Cloud Platform. Cada empresa ofrece un coste-beneficio particular y se especializa en uno u otro servicio relacionado a la

computación, por lo que, para resumirlas, se ha optado por rescatar las características más importantes de cada proveedor y detallarlos en la **Tabla 3-2**.

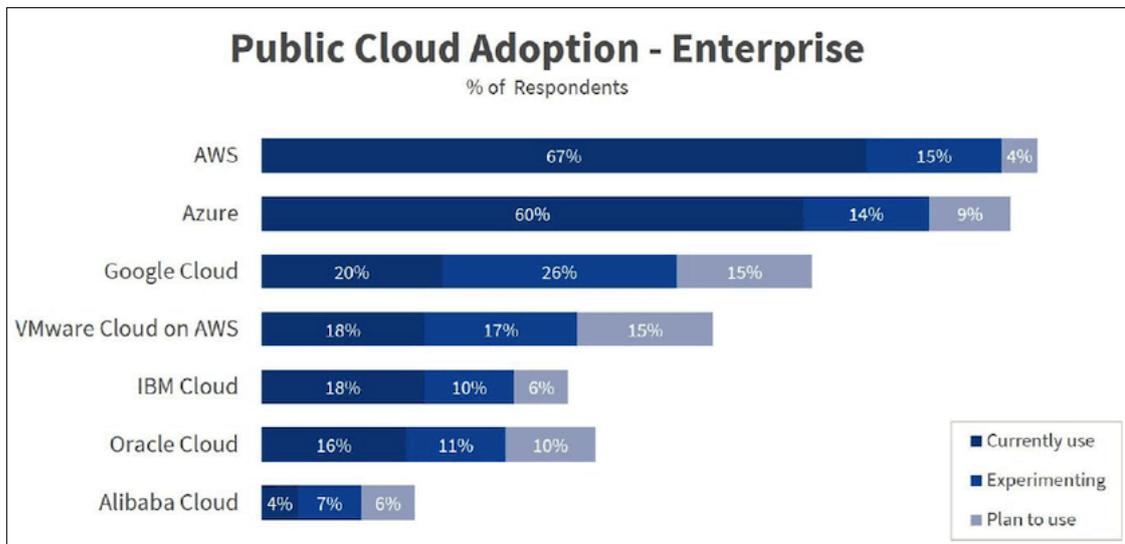


Ilustración 4-2: Adopción de plataformas Cloud en empresas

Fuente: (Flexera, 2019)

Tabla 3-2: Herramientas en el mercado del Cloud Computing

Característica	Amazon Web Service	Microsoft Azure	Plataforma Google Cloud
Tecnologías de red	Servicios VPC Amazon Route 53 Cloud Front Conexión AWS Direct	Caché de rama Servicio DNS Equilibrio de carga	Interconexión en la nube VPN en la nube Peering del operador y directo
Tecnologías de almacenamiento	Servicio de almacenamiento simple Amazon Glacier Amazon Elastic Book Store Amazon Elastic File System	Almacenamiento de objetos escalable datos de texto y binarios. Azure Files Azure Queues	Blob Store de Google Cloud SQL
Soporte para bases de datos relacionales	Amazon-RDS SQL MySQL Oracle	Azure SQL Azure SQL-DW	Cloud-SQL Cloud-Spanner
Tipo de Modelo de Servicio	IaaS, PaaS, SaaS. Principalmente IaaS	IaaS, PaaS, SaaS. Principalmente SaaS	IaaS, PaaS, SaaS. Principalmente PaaS.
Enfoque	Ofertas para empresas y nube pública.	Integración de herramientas de Microsoft, código abierto y nube híbrida.	Código abierto y portabilidad y nube híbrida.
Disponibilidad de regiones y gama de servicios	66 regiones	54 regiones	20 regiones, aunque se puede ser expandible
Seguridad	Moderada	Moderada	Moderada

Naturaleza de los servicios	Proporciona servicios centrados en la empresa	No está totalmente “preparado para la empresa”	Diseñado para negocios basados en la nube
Mayor inconveniente	Difícil de usar, las opciones pueden ser confusas	Las herramientas de gestión son ineficientes	Cuenta con centros de datos limitados en todo el mundo
Precio y pago	Gratis Comercial (Pay-as-you-go): \$0.11/hora Suscripción: Por horas o mensual	Comercial (Pay-as-you-go): \$0.02/minuto Suscripción: Mensual	Gratis Comercial (Pay-as-you-go): \$0.02/minuto Suscripción: Por horas o mensual

Fuente: (Saraswat, M.; Tripathi R., 2020)

Realizado por: Cuenca A.; Llinin F., 2022.

2.2.5. Implementación de software tipo SaaS

(Núñez, 2014) en su trabajo menciona que para implementar un SaaS se debe ver un modelo de despliegue de la infraestructura definida por el objetivo de la nube y la naturaleza en que la nube se encuentra. Para la implementación de un sistema bajo el modelo SaaS se debe seguir un debido proceso como se ve en la **Ilustración 5-2** y en cada una de las etapas se realizan diferentes tareas para cumplir con la finalización de un producto software óptimo. Asimismo, se destacan que los pasos a seguir son los requerimientos, seguido del análisis del proceso del negocio. En la etapa del diseño la tarea más importante a destacar es la elección de la arquitectura del servicio. Finalmente se realiza la implementación y codificación del sistema, el mismo que para su validación deberá someterse a pruebas para su aprobación técnica.



Ilustración 5-2: Proceso de implementación de aplicaciones SaaS

Fuente: (Software Gurú, 2014)

2.3. Herramientas de desarrollo

En esta sección se brinda una breve definición teórica de las herramientas (stack) que se van a ocupar en el desarrollo del sistema. Así pues, se repasan los lenguajes de programación, frameworks web, bases de datos, librerías y/o módulos y entornos de desarrollo inmiscuidos en los requerimientos que precisa el sistema.

2.3.1. Lenguajes de Programación

2.3.1.1. Python

Python es un lenguaje de programación que está orientado a objetos y que, gracias a su estructuración, es claro y muy potente. Así pues, el lenguaje cuenta con una sintaxis simple y elegante que facilita la creación y lectura de los programas que se escriben. Estas características lo convierten en el lenguaje ideal para el desarrollo de prototipos y tareas de programación ad hoc. De igual forma, ofrece un conjunto de datos que, para una mejor comprensión, se han resumido en la **Tabla 4-2**.

Tabla 4-2: Tipos de datos en Python

Nombre	Tipo	Descripción
Enteros	int	Números con únicamente parte entera, tales como: 5, 423, 3271
Punto Flotante	float	Números con punto decimal, tales como: 2.3, 4.2, 4.212
Cadenas	str	Secuencia ordenada de caracteres, tales como: "hola", "Alex", "2000", "#/("
Listas	list	Secuencia ordenada de objetos, tales como: [10, "hola", 200.3]
Diccionarios	dict	Valores en pares con clave desordenada, tales como: {"nombre": "alex", "apellido": "cuenca"}
Tuplas	tup	Secuencia de objetos de orden inmutable, tales como: (10, "hola", 200.3)
Conjuntos	set	Colección desordenada de objetos único, tales como: {"a", "b"}
Boléanos	bool	Valor lógico que indica Verdad (True) o Falsedad (False)

Fuente: (Ren, S., 2019)

Realizado por: Cuenca A.; Llinin F., 2022.

2.3.1.2. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con

JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Eguiluz, 2009).

2.3.2. Frameworks Web

2.3.2.1. ReactJS

Es una biblioteca de código abierto que se utiliza para crear interfaces de usuario explícitamente para aplicaciones de una sola página. Permite a los ingenieros de software crear una aplicación web inmensa que puede usar datos y puede cambiar después de un tiempo sin recargar la página (Rawat & Mahajan, 2020). Está basada en componentes que se implementan para el desarrollo de interfaces de usuario interactivas. Actualmente es la biblioteca JS front-end más popular incorporando un modelo MVC (Aggarwal, 2018).

2.3.2.2. Django

Es un framework para el desarrollo web usando Python y creado para el desarrollo rápido de sitios basados en bases de datos. Django utiliza una variación del modelo MVC, el paradigma MTV. Las comparaciones de ambos patrones pueden ser observadas en la **Tabla 5-2**. Django permite, de manera predeterminada, utilizar una capa de mapeo relacional de objetos (ORM). Además, en un proyecto Django se pueden incluir plantillas con extensión HTML para la visualización de datos y para operaciones CRUD.

Tabla 5-2: Relación entre el modelo MVC y MVT de Django

Modelo MVC tradicional	Modelo MVC y archivos de Django
Modelo	Modelo (models.py)
Vista	Plantilla (template.html)
Controlador	Vista (views.py)

Fuente: (Vainikka, J., 2018)

Realizado por: Cuenca A.; Llinin F., 2022.

2.3.3. Base de datos

2.3.3.1. PostgreSQL

PostgreSQL es un sistema de base de datos relacional que utiliza el lenguaje SQL para la carga y almacenamiento de datos en una aplicación. Del mismo modo, cuenta con una arquitectura

probada, fiabilidad, integridad de datos, robustez, extensibilidad y una comunidad muy grande de desarrolladores que brindan soporte activo a los usuarios (PostgreSQL, 2022).

2.3.4. Librerías y/o módulos

2.3.4.1. Tailwind

Es un marco CSS de bajo nivel basado en utilidades altamente personalizable que se puede usar para construir rápidamente la interfaz de usuario. No tiene componentes prefabricados, pero tiene amplios recursos de clases y conjuntos de utilidades que permiten a las personas crear cualquier diseño que desee. Esa es la razón por la que todos los componentes son fáciles de personalizar y administrar (Tailwind, 2019).

2.3.4.2. Semantic UI

Semantic UI es un marco de trabajo de desarrollo que ayuda en la creación de hermosos diseños que además de ser responsivos utilizan sintaxis HTML amigable con el programador (Semantic UI, 2018). Utiliza frases sencillas que permiten ejecutar acciones propias de JavaScript y cada una de estas acciones son fácilmente modificables por parte de los desarrolladores (React, 2022).

2.3.5. Entorno de desarrollo

2.3.5.1. Visual Studio Code

Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La selección principal para desarrolladores web y JavaScript, con multitud de extensiones para admitir casi cualquier lenguaje de programación (MICROSOFT, 2021).

2.4. Metodología de desarrollo Kanban

Según (Bermejo, Florit y Sedó, 2011), Kanban se deriva de dos palabras japonesas, kan, que significa “visual”, y ban, “tarjeta”, la misma que se denomina como una metodología de producción del trabajo basada en señales visuales para la gestión del esfuerzo y dedicación del equipo de producción. Como se observa en la **Ilustración 6-2**, esta metodología tiene como punto central determinar tareas por hacer y cambiar su prioridad en función de los acontecimientos. Además, la cadena de trabajo está visible para todos, y en el caso de existir problemas es mucho más fácil conocer lo que se está produciendo (Banco Bilbao Vizcaya Argentaria S.A., 2019).

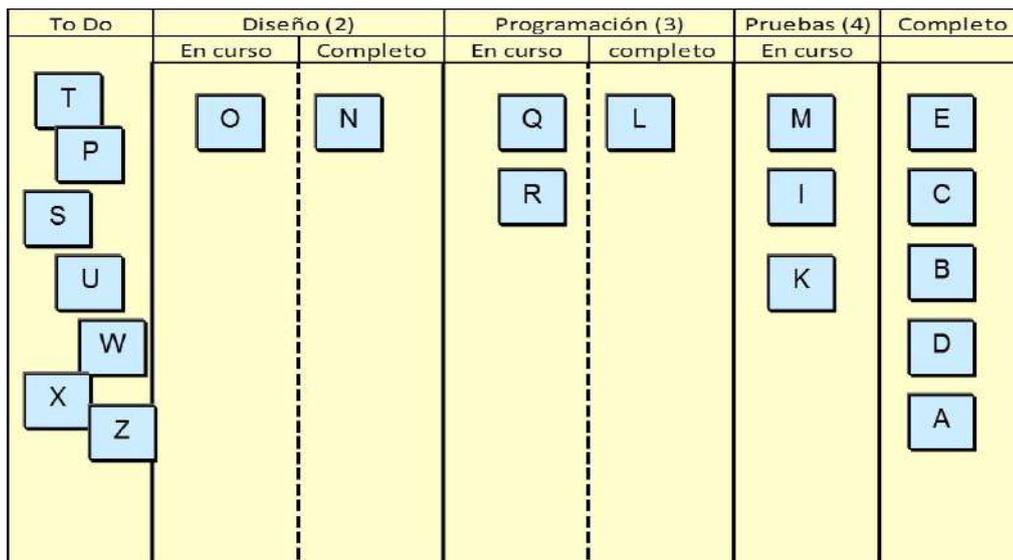


Ilustración 6-2: Metodología Kanban

Fuente: (Stackscale, 2020)

De acuerdo con (Pérez 2012) existen 3 reglas principales para la metodología Kanban, las cuales se indican en la **Ilustración 7-2**.

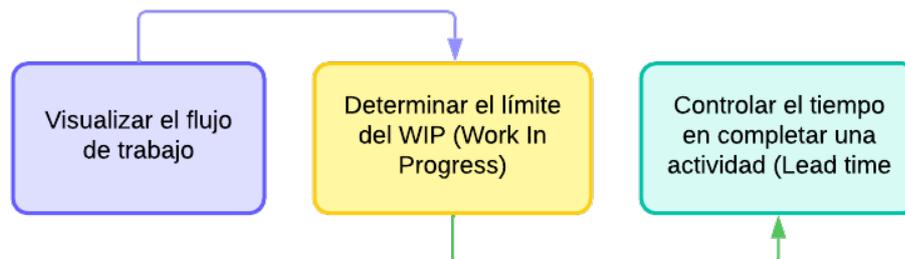


Ilustración 7-2: Reglas principales de Kanban

Fuente: (Pérez, 2016)

La regla de visualizar el flujo de trabajo hace referencia a dividir el trabajo en tareas, escribirlas en una tarjeta y colocarlas en la columna inicial; las columnas pueden ser tantas como el equipo considere necesario. Con respecto al WIP se busca limitar el número de tareas que se pueda realizar en cada estado del flujo de trabajo. Finalmente, el lead time hace referencia al tiempo transcurrido desde que inicia su petición hasta su entrega, es decir mide el rendimiento del proceso.

Esta metodología se basa en el uso del tablero Kanban que según (Llerena & Guillen, 2020) es aquel artefacto que proporciona visibilidad de todo el proceso de software, mostrando el trabajo asignado a cada miembro del equipo. Además, indica las prioridades de cada tarea y resalta los cuellos de botella existentes, de este modo el equipo se concentra en los problemas que se presentan en el proceso buscando una solución rápida, minimizando los defectos y manteniendo un flujo estable.

2.4.1. Roles de la metodología Kanban

Bajo la mentalidad general de Kanban “menos es más” y acorde con (Roche, 2020) Kanban no establece ningún rol en absoluto, pero esto no quiere decir que se puedan crear de forma libre en caso de ser necesario sobre todo si se tratan de proyectos muy grandes, además que es libre de añadir otros roles siempre y cuando añadan valor y no generen conflictos con otros elementos del proyecto desarrollado.

2.4.2. Implementación

De acuerdo con (Lendínez, 2019) para la implementación correcta del sistema Kanban será necesario seguir una serie de pasos como se muestra en la **Ilustración 8-2** :

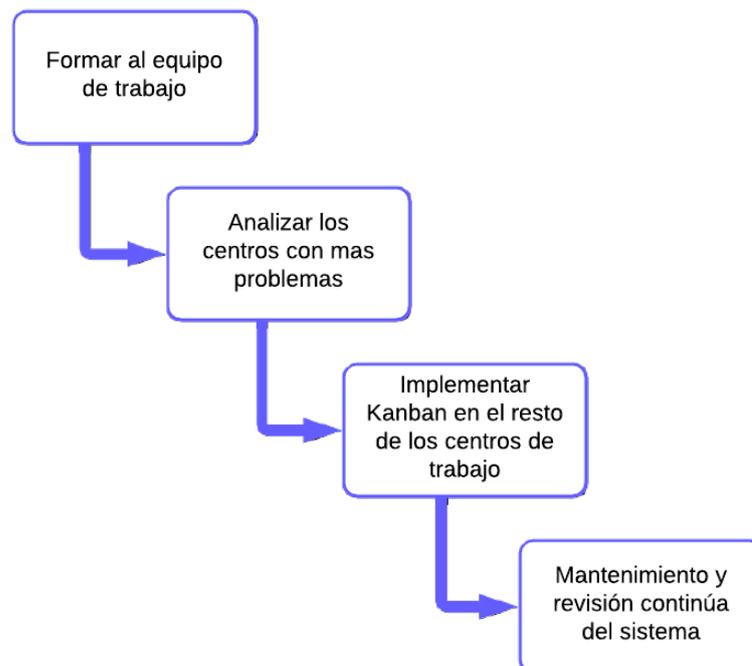


Ilustración 8-2: Flujo de trabajo principal de Kanban

Fuente: (Stackscale, 2020)

En el primer paso de **la Ilustración 8-2** se forma el equipo de trabajo de Kanban teniendo en cuenta los beneficios y ventajas que presenta el uso la metodología. El siguiente paso es realizar un análisis previo de los centros con más problemas para identificar los posibles fallos. A continuación, se debe implementar Kanban en el resto de los centros de trabajo para cumplir con las metas en el desarrollo. Y, finalmente se debe aplicar mantenimiento y revisión continúa del sistema. Debido a la flexibilidad de la metodología, en este flujo de trabajo no es necesario implementar Kanban en todas las fases detalladas.

2.5. Norma ISO/IEC 25010: Modelos de calidad de sistemas y software

La norma ISO/IEC 25010 proviene de la familia ISO/IEC 25000:2005 - SQuaRE (Requisitos y evaluación de la calidad del producto de software), cuyo objetivo es la creación de un marco de trabajo común para evaluar la calidad del producto de software. Esta familia surge como el resultado de la evolución de normas anteriores como la norma ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto de software (ISO/IEC 25000, 2005).

La norma ISO/IEC 25010 determina las características de calidad que evalúan las propiedades de un producto de software desarrollado. Se puede interpretar como el grado en que dicho software satisface los requerimientos para los que fue construido.

2.5.1. Características de la ISO/IEC 25010: 2011

La **Tabla 6-2** detalla algunas de las características de esta norma.

Tabla 6-2: Características y subcaracterísticas de la ISO/IEC 25010 2011

Característica	Definición	Subcaracterísticas
Adecuación funcional	Capacidad que posee el producto software para satisfacer las necesidades del usuario, cuando se usa en condiciones específicas.	<ul style="list-style-type: none"> • Completitud funcional • Corrección funcional • Pertinencia funcional
Eficiencia de desempeño	Capacidad del producto software para proporcionar un buen desempeño en relación con los recursos del sistema.	<ul style="list-style-type: none"> • Comportamiento temporal • Utilización de los recursos • Capacidad
Compatibilidad	Capacidad del sistema o componente para intercambiar información con otro producto software cuando comparten el mismo entorno de hardware o software.	<ul style="list-style-type: none"> • Coexistencia • Interoperabilidad
Usabilidad	Grado en que el producto software puede ser usado, aprendido, entendido y resultar cómodo para el usuario.	<ul style="list-style-type: none"> • Adecuación • Aprendizaje • Operabilidad • Protección contra errores del usuario

		<ul style="list-style-type: none"> • Estética • Accesibilidad
Fiabilidad	Capacidad del producto software para desempeñar funciones específicas, en condiciones y periodo de tiempo determinados.	<ul style="list-style-type: none"> • Madurez • Disponibilidad • Tolerancia a fallos • Capacidad de recuperación
Seguridad	Grado en que un producto software mantiene la confidencialidad, integridad y accesibilidad de la información y los datos, de personas o sistemas no autorizados.	<ul style="list-style-type: none"> • Confidencialidad • Integridad • No repudio • Responsabilidad • Autenticidad
Mantenibilidad	Capacidad en que un producto software puede ser modificado de manera eficaz y eficiente para mejorar, corregir o adaptar a nuevos requerimientos.	<ul style="list-style-type: none"> • Modularidad • Reutilización • Analizabilidad • Capacidad de ser modificado • Capacidad para ser probado
Portabilidad	Capacidad del producto software para ser trasladado de forma efectiva de uno entorno hardware o software a otro.	<ul style="list-style-type: none"> • Adaptabilidad • Facilidad de instalación • Capacidad para ser reemplazado

Fuente: (ISO/IEC 25010, 2011)

Realizado por: Cuenca A.; Llinin F., 2022.

El presente trabajo se centra en la característica “Eficiencia de desempeño” y las dos primeras subcaracterísticas que lo componen.

2.5.2. Eficiencia de desempeño

De acuerdo con la (ISO/IEC 25010, 2011) esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas como se observa en la **Tabla 7-2**.

Tabla 7-2: Subcaracterísticas de eficiencia de desempeño

Comportamiento temporal	Grado en que los tiempos de respuestas y procesamiento de un producto software al desempeñar sus funciones en condiciones específicas cumple con los requisitos establecidos.
Utilización de recursos	Grado en que los recursos y tipos de recursos utilizado por un producto software cuando desempeña sus funciones cumplen con los requisitos.
Capacidad	Grado en que un producto software cumple con los requisitos cuando se toman en cuenta los límites máximos de sus parámetros.
Comportamiento temporal	Grado en que los tiempos de respuestas y procesamiento de un producto software al desempeñar sus funciones en condiciones específicas cumple con los requisitos establecidos.

Fuente: (ISO/IEC 25010, 2011)

Realizado por: Cuenca A.; Llinin F., 2022.

2.6. Trabajos relacionados

Después de realizar una revisión bibliográfica de varios de los trabajos relacionados al tema a desarrollar, se destacan los siguientes:

El trabajo de (Chicaiza, 2020) “Desarrollo de un sistema vía web para automatizar los pedidos y reservaciones de comidas en la parrilla “Don Carlos” del cantón Milagro”, implementa un sistema para la automatización de los procesos en el restaurante con el fin de tener control en las órdenes que llegan a cocina, mejorando así la rapidez en el servicio. Este trabajo se desarrolló usando la metodología XP, donde utilizó el framework web Django, además, como motor de base de datos se emplea PostgreSQL y lo cuelga en el servidor Ubuntu 18.04. Como resultado del trabajo el sistema cumple con los parámetros requeridos para realizar reservación y pedidos, gestionar al cliente con número de turnos, siendo el sistema fácil de usar y seguro.

El trabajo de (Mendoza, 2019) titulado “Desarrollo de un sistema web, que permita administrar pedidos, aplicada a la distribuidora repuestos Córdoba” identifica las actividades manuales que provocan demora en el proceso. Con el sistema web desarrollado se vio una mejora en la eficiencia en el proceso de gestión de pedidos. Este trabajo se desarrolló usando la metodología SCRUM en donde usó el framework Laravel y como motor de base de datos emplea PostgreSQL. Como resultado final el sistema cumple con los parámetros requeridos para realizar la administración de pedidos.

(Llerena y Guillen, 2020) en su trabajo “Aplicación de la metodología Kanban en el desarrollo del software para generación, validación y actualización de reactivos, integrado al sistema informático de control académico UNACH”, usó la metodología Kanban para el desarrollo del software propuesto. Además, este se integra al sistema SICOA de la UNACH. Este trabajo se realizó utilizando la arquitectura orientada a dominio y la tecnología ASP.Net Core.

CAPITULO III

3. MARCO METODOLÓGICO

El siguiente capítulo precisa la metodología seguida para la elaboración de la solución propuesta para el Trabajo de Integración Curricular que tiene por nombre “My iCard”.

3.1. Tipo de estudio

El presente Trabajo de Integración Curricular será de tipo Aplicativo, debido a que se va a desarrollar un sistema web bajo el modelo SaaS para la automatización del proceso actual de la pizzería aplicando los conocimientos que se han obtenido a lo largo de la carrera.

3.1.1. Métodos y técnicas

De acuerdo con los objetivos específicos planteados en este trabajo en la **Tabla 1-3** se detallan los métodos, técnicas y fuentes empleadas para el desarrollo del trabajo.

Tabla 1-3: Métodos, Técnicas y Fuentes para el Trabajo de Integración Curricular

Objetivos	Métodos	Técnicas	Fuentes
Establecer el proceso que lleva actualmente el establecimiento para manejar los pedidos en época post pandemia.	Observación Analítico	<ul style="list-style-type: none">• Observación• Entrevista• Diagrama Conceptual BPMN	<ul style="list-style-type: none">• Meseros de la pizzería• Gerente de la pizzería
Investigar sobre características, beneficios, desventajas y proceso de implementación que conlleva aplicar el modelo SaaS en el desarrollo de software.	Analítico	<ul style="list-style-type: none">• Revisión de documentación	<ul style="list-style-type: none">• Internet• Bases de datos documentales• Revistas• Congresos
Desarrollar los módulos de Autenticación, Categorías, Mesas, Productos, Usuarios, Pedidos e Historial de Pagos.	KANBAN Analítico	<ul style="list-style-type: none">• Tablero Kanban• Tarjetas Kanban• Reuniones periódicas	<ul style="list-style-type: none">• Visual Studio Code• Usuarios expertos• Internet• Trello

Evaluar la eficiencia del sistema web mediante la aplicación del estándar ISO/IEC 25010.	Inductivo Observación	<ul style="list-style-type: none"> • Observación • Análisis estático y dinámico del código 	<ul style="list-style-type: none"> • Usuarios que usaran el software • JMeter • Sistema My iCard
--	--------------------------	--	---

Realizado por: Cuenca A.; Llinin F., 2022.

3.2. Gestión del proyecto

3.2.1. Análisis previo al desarrollo del proyecto

En esta sección se lleva a cabo una investigación previa al desarrollo del proyecto de integración curricular. De esta manera se puede juzgar que tan factible resultará su realización, identificar los inminentes riesgos que pueden aparecer antes, durante y después del desarrollo. Así, se prevé evitar la mayor cantidad de inconvenientes en cualquier fase del proyecto.

3.2.2. Proceso del negocio de la pizzería

Como resultado de las reuniones de trabajo llevadas a cabo con el administrador y personal de la pizzería, se identificó el modelo del negocio existente y el que se aspira a tener. Se elaboró una primera representación del proceso de toma de pedidos como se observa en la **Ilustración 1-3**, la que posteriormente se llevó a la notación BPMN 2.0 de Bizagi de la **Ilustración 2-3**. Como consecuencia del análisis efectuado, se plantea el diagrama SIPOC como el de **Tabla 2-3** que permite identificar los involucrados en el proceso de atención al cliente.

Tabla 2-3: Diagrama SIPOC del proceso de toma de pedidos

S	I	P	O	C
Cliente	Menú, selección	Registro pedido	Pedido registrado	Cocinero
Cocinero	Pedido Registrado	Elaborar pedido	Pedido elaborado	Mesero
Mesero	Pedido Elaborado	Servir pedido	Pedido servido	Cliente
Cliente	Pedido servido, solicitud de cuenta	Cobrar pedido	Pedido cobrado, valor cuenta	Cajero, Cliente
Cajero, Cliente	Pedido cobrado, cuenta	Pagar pedido	Pedido pagado, comprobante, vuelto	Cliente

Realizado por: Cuenca A.; Llinin F., 2022.

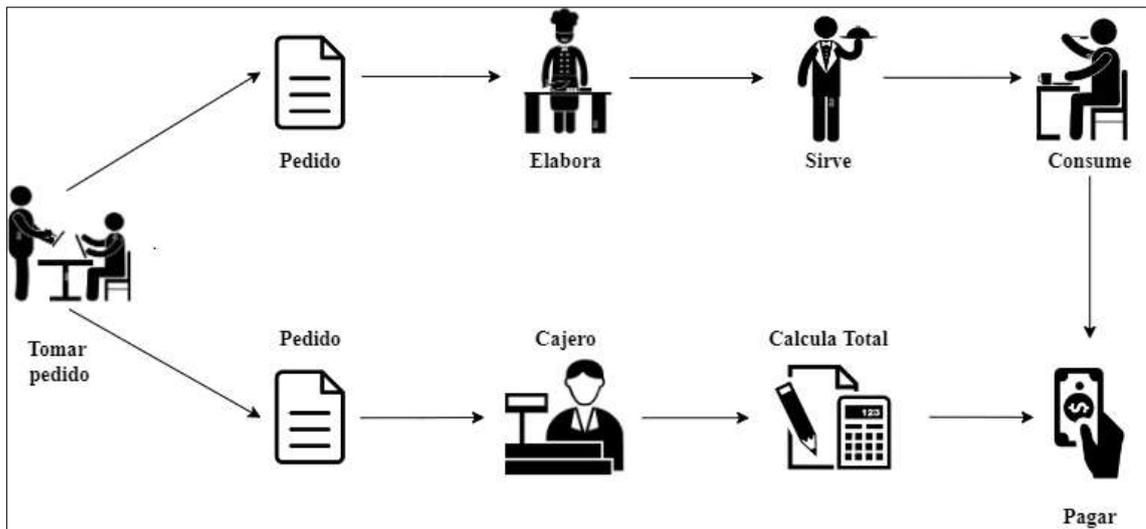


Ilustración 1-3: Proceso actual de toma de pedidos en la pizzería

Realizado por: Cuenca A.; Llinin F., 2022.

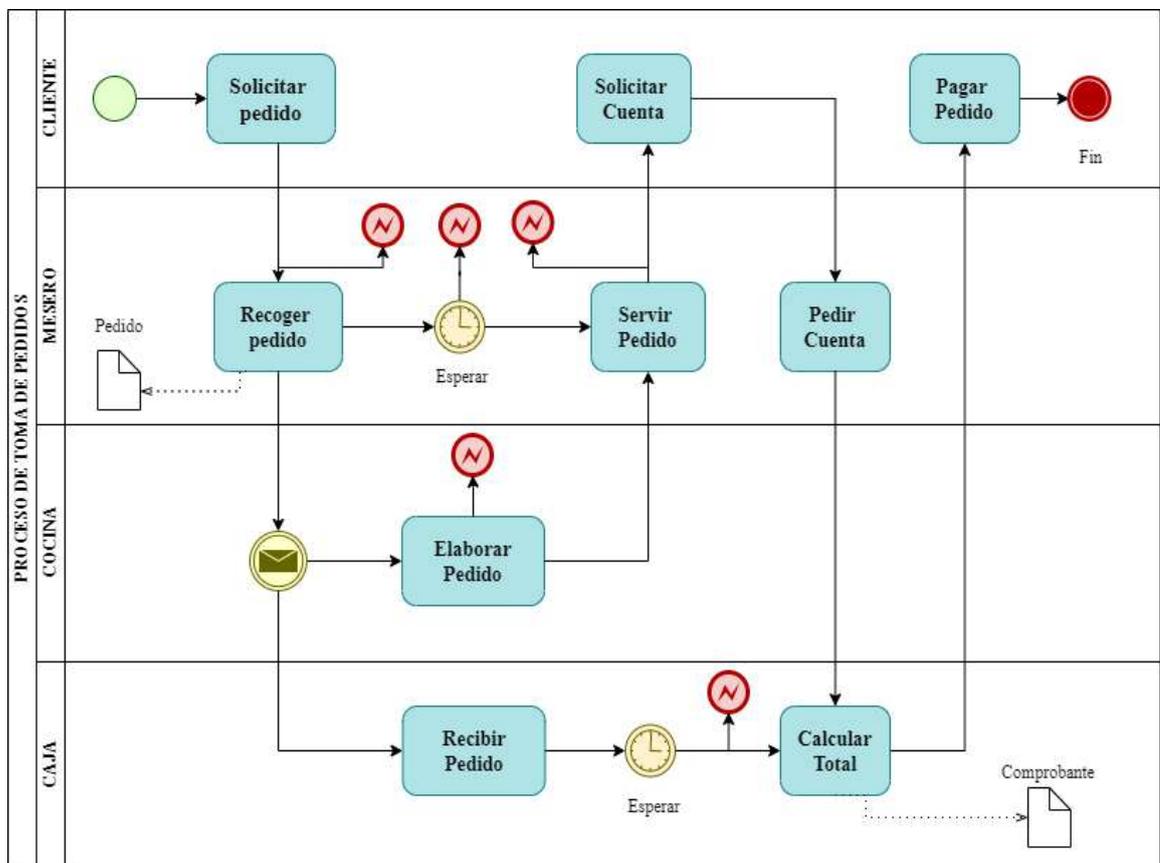


Ilustración 2-3: Modelo actual del proceso de toma de pedidos actual en BPMN

Realizado por: Cuenca A.; Llinin F., 2022.

En consideración a las debilidades encontradas, junto con el análisis de las buenas prácticas imperantes en los restaurantes y las tecnologías disponibles, se rediseñó el proceso sobre la base

de mejorar la eficiencia en lo que respecta a la toma de pedidos en el contexto de postpandemia. La representación y el modelamiento de este rediseño se muestran en la **Ilustración 3-3** e **Ilustración 4-3**.

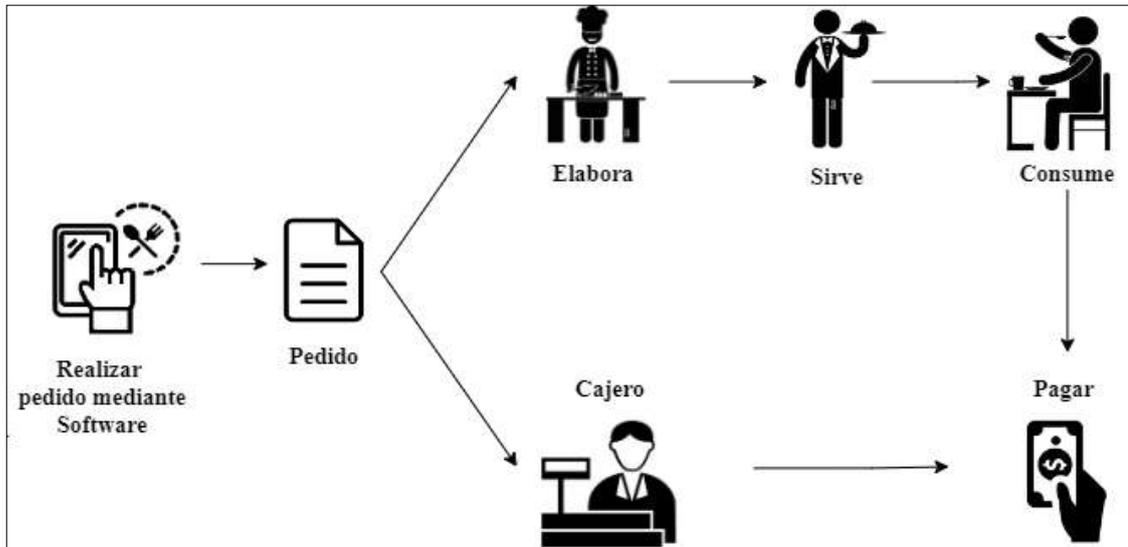


Ilustración 3-3: Proceso de toma de pedidos deseado en la pizzería

Realizado por: Cuenca A.; Llinin F., 2022.

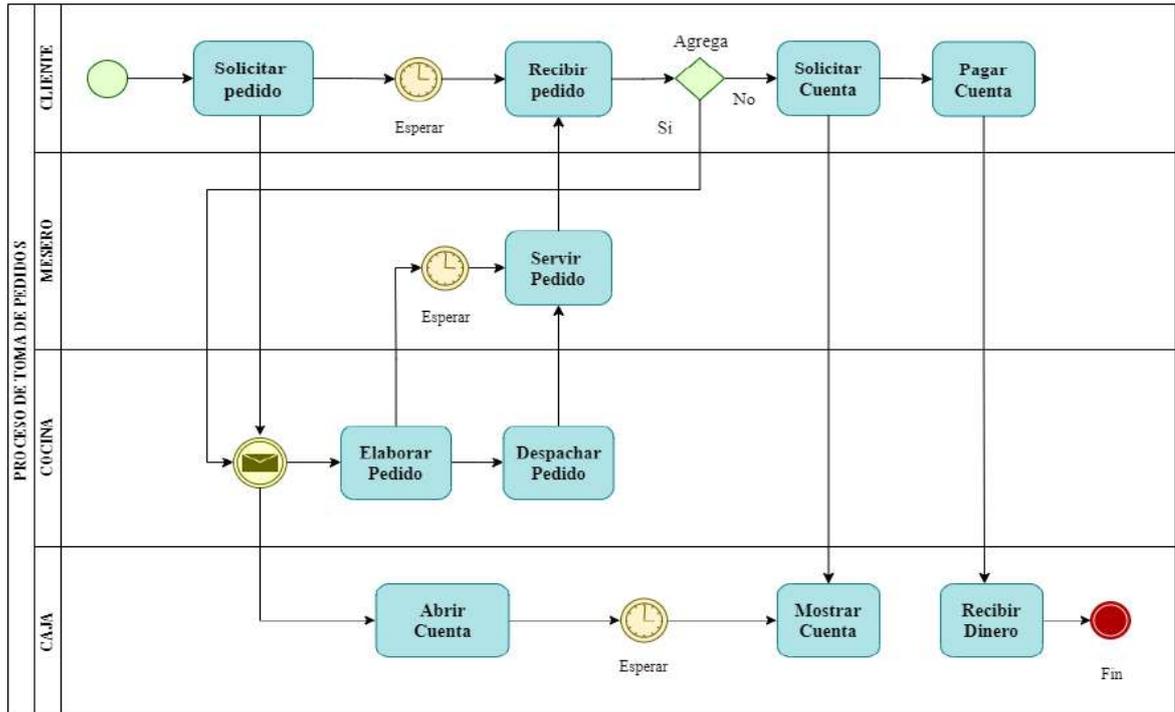


Ilustración 4-3: Modelo propuesto para mejorar la eficiencia en el proceso de toma de pedidos en BPMN

Realizado por: Cuenca A.; Llinin F., 2022.

3.2.3. Estudio de la factibilidad

Se efectúa el presente estudio con el objetivo de conocer la inversión que se requiere para el desarrollo, la viabilidad y disponibilidad de los recursos humanos, materiales y financieros del sistema de “My iCard”.

De acuerdo con el análisis del hardware y software por el equipo de desarrollo, se determina que el mismo dispone de los recursos necesarios para la ejecución del proyecto. Con el análisis de costo-beneficio para la implementación del mismo, se determina que al cabo de un mes se recupera la inversión. Actualmente en el Ecuador no existe ningún impedimento legal que permita a una empresa adquirir una licencia de soluciones AWS que es en donde se desplegará la aplicación, en referencia a la estimación COCOMO para los recursos humanos y el esfuerzo necesario para la ejecución de este proyecto indica que el esfuerzo es de 7.9183534 hombres-mes, y se necesitan dos programadores para el desarrollo del sistema. Para el desarrollo se estima una duración de cinco meses desde el inicio del proyecto.

Por lo anteriormente expuesto se concluye que el proyecto es 80% viable para la ejecución completa. Las estimaciones realizadas se encuentran en el Manual técnico entregado al propietario del establecimiento como se lo observa en el **Anexo A**.

3.2.4. Gestión de riesgos

Con el fin de identificar los potenciales problemas y/o riesgos se presenta un plan para examinarlos y abordarlos de manera que, en caso de aparecer, impacten mínimamente en el desarrollo del proyecto. Para la gestión de riesgos se determinan los factores potenciales de riesgo que ocasionarían mayor impacto durante el desarrollo de My iCard como se lo observa en el Manual Técnico.

Como se observa en la **Tabla 3-3**, se realizó el tratamiento de la prevención e incurrancia de los riesgos detallados en el estudio para evitar retrasos en el desarrollo del proyecto; la gestión como tal, se la maneja a través del siguiente formato:

Tabla 3-3: Hoja de gestión de riesgo

HOJA DE GESTIÓN DE RIESGO		
ID. DEL RIESGO: R01		Fecha: 2022-05-22
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Prioridad: 1
DESCRIPCIÓN: La inexperiencia por parte de los desarrolladores en cuanto a las herramientas a utilizar pueden ocasionar un gran retraso.		
REFINAMIENTO: <ul style="list-style-type: none">• CAUSAS:<ul style="list-style-type: none">○ Durante el periodo estudiantil se ha adquirido conocimientos básicos en cuanto a programación.○ Falta de colaboración por parte del equipo de trabajo.• CONSECUENCIAS:<ul style="list-style-type: none">○ Extensión de tiempo en el desarrollo del proyecto.○ Incertidumbre en cuanto a el resultado a obtener por la falta de conocimientos.		
REDUCCIÓN: <ul style="list-style-type: none">• Realizar un autoanálisis al equipo de desarrolladores para identificar a tiempo las falencias en cuanto a conocimientos para de esta manera reducir el riesgo.		
SUPERVISIÓN: <ul style="list-style-type: none">• Estar al tanto del progreso del proyecto.• Realizar un estudio permanente de los temas que falte enriquecer el conocimiento.		
GESTIÓN: <ul style="list-style-type: none">• Dedicar tiempo a realizar un estudio de las herramientas que se va a utilizar.		

Realizado por: Cuenca A.; Llinin F., 2022.

3.3. Ingeniería del Producto

3.3.1. Metodología Kanban para el desarrollo del software

La metodología que se empleó en el análisis e implementación del producto de software fue KANBAN. Teniendo en cuenta que el enfoque evolutivo de Kanban hace que la resistencia del equipo y de las partes interesadas sea escasa o nula (Digité, 2021), y recurriendo al flujo de trabajo de Kanban, las revisiones estratégicas (Strategy review) y reuniones de entrega de servicios (Service delivery meeting) se realizan en conjunto con el cliente con el fin de preservar su satisfacción mediante entregas eficiente y de alta calidad.

3.3.2. Backlog del producto (Product backlog)

En la metodología Kanban el Product Backlog es una representación visual de los elementos que se pueden o no puede entregar como parte de un entregable al cliente. En un tablero Kanban, el backlog se halla mucho antes de las tareas por hacer, por lo que toma un papel muy importante en el desarrollo del producto pues marca el punto de partida de casi todas las actividades que se llevarán a cabo.

Dicho esto, hay que señalar que el propósito del backlog Kanban no es sólo enumerar ideas o comentarios de los clientes. Por el contrario, su intención debe ser hacerlos continuamente más procesables para finalmente darles vida.

En la **Ilustración 5-3** se puede observar un ejemplo del Product Backlog del proyecto. El detalle de las ideas previas al desarrollo y cada una las posibles actividades que se deberán realizar para completar la gestión y la ingeniería del trabajo. La evolución mensual del tablero se lo puede observar en el **Anexo B**.

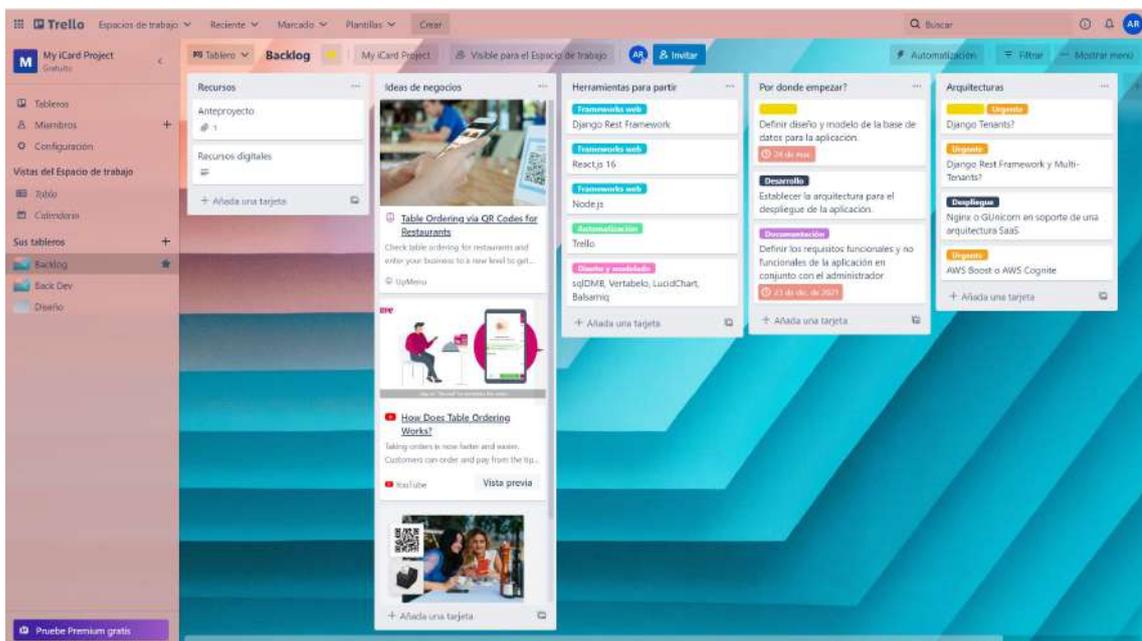


Ilustración 5-3: Tablero Kanban del Backlog de “My iCard”

Realizado por: Cuenca, A.; Llinin, F., 2022.

3.3.3. *Requerimientos*

Todo requisito aquí especificado describirá comportamientos externos del sistema, visibles por parte de los usuarios, operadores y otros sistemas. Esta es sección se la puede encontrar en el Anexo F del Manual Técnico.

En la **Tabla 4-3**, se pueden encontrar los requisitos funcionales del sistema.

Tabla 4-3: Requisitos funcionales

ID	REQUISITOS
RF-1	El sistema debe permitir al cliente realizar pedidos.
RF-2	El sistema debe permitir al cliente consultar los productos consumidos, visualizar el costo de cada uno de ellos y el total, del pedido
RF-3	El sistema debe permitir al cliente modificar un pedido antes de confirmar
RF-4	El sistema debe permitir al administrador anular un pedido
RF-5	El sistema debe permitir al administrador modificar un pedido.
RF-6	El sistema debe permitir al administrador consultar las mesas.
RF-7	El sistema debe permitir al administrador cambiar el estado de un pedido a: abierto o cerrado
RF-8	El sistema debe permitir al administrador cambiar el estado de un producto del pedido
RF-9	El sistema debe permitir al administrador obtener informe de productos vendidos
RF-10	El sistema debe permitir al administrador consultar los ingresos del restaurante mediante un historial de pagos.
RF-11	El sistema debe permitir al administrador mantener los productos
RF-12	El sistema debe permitir al administrador mantener las categorías de los productos
RF-13	El sistema debe permitir al administrador mantener las mesas
RF-14	El sistema debe permitir al administrador consultar el pedido de un cliente
RF-15	El sistema debe permitir al administrador generar una boleta de pago del cliente
RF-16	El sistema debe permitir al administrador consultar las existencias de un producto

Realizado por: Cuenca A.; Llinin F., 2022.

3.3.4. Diseño

Con la información recogida en la etapa de requerimientos se puede establecer una estructura que satisfaga los requisitos de la parte interesada. Así también, se fija la arquitectura mediante la cual las distintas capas del producto interactuarán.

3.3.4.1. Arquitectura del sistema

Debido a que la presente solución tiene como objetivo implementar un software tipo SaaS se debe contar un back-end capaz de soportar una arquitectura Multi-Tenant para que de esta forma la creación y el alojamiento de instancias sea posible en la fase de despliegue. Teniendo en cuenta lo anterior se detalla la arquitectura Multi-Tenant.

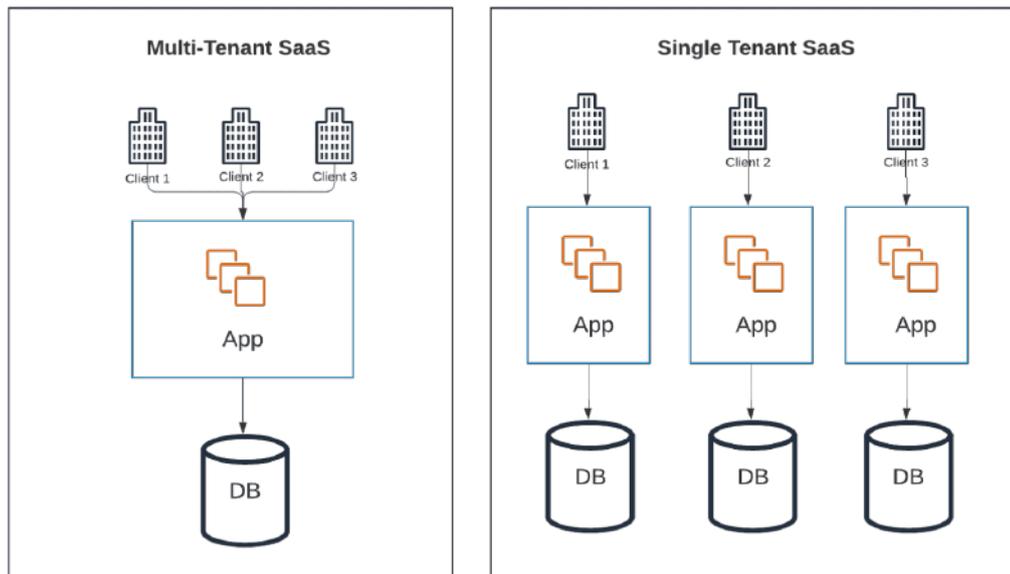


Ilustración 6-3: Arquitectura Multi-Tenant y Single-Tenant

Fuente: (Medium, 2021)

A diferencia de un modelo de inquilino único, la arquitectura multiarrendatario (Multi-Tenant) permite alojar y servir a varios usuarios en un único entorno. La infraestructura que se ofrece a los clientes es enteramente compartida, aislada entre organizaciones y con servicios que están centralizados en la necesidad del cliente. Para el sistema “My iCard” se establece el modelo pool de dominios para el establecimiento de direcciones únicas de cada cliente. De esta forma, se asegura un compartimento de infraestructura y recursos manteniendo los datos de cada uno de los clientes completamente aislados e invisibles para los demás.

3.3.4.2. Base de datos

Construir un base de datos con un buen diseño ayuda a mejorar el rendimiento, coherencia, precisión y fiabilidad de los datos. Al brindar prioridad a estos conceptos se garantiza una mejora en el rendimiento de acceso a los datos y la reducción del espacio que ocupan estos. En consecuencia, se puede construir una base de datos que sea fácil de usar y mantener.

a) Diseño conceptual

En esta etapa del diseño se pretende construir un esquema conceptual donde se presenten las entidades, relaciones y atributos que intervengan en el proceso a automatizar. Teniendo en

cuenta los puntos anteriores y siguiendo las buenas prácticas del diseño conceptual, se han definido las siguientes etapas:

i) Identificación de entidades

Luego de sentar los flujos y las reglas de negocio para el sistema informático iCard, se han planteado las siguientes entidades:

- Cliente
- Administrador
- Mesa
- Pedido
- Categoría
- Alimento
- Pago

Para las dos primeras entidades en la lista anterior es preciso aclarar que el framework Django usado para realizar la aplicación se maneja el concepto de AbstractUser. El concepto de usuario abstracto maneja a los usuarios con un solo modelo y para establecer roles (administrador, cliente, etc.) se establecen atributos que varíen dependiendo de los privilegios que se le quiera otorgar al usuario al momento de registrarlo.

ii) Identificar y asociar atributos con entidades o relaciones

Al igual que en la etapa de Identificación de entidades, se ha recurrido al análisis de los flujos y reglas de negocio. En este sentido, se establecen los atributos tentativos para cada una de las entidades que participarán en el sistema My iCard.

iii) Determinar los dominios de los atributos

Habiendo determinado los atributos para las entidades, en esta fase se establecen el conjunto de valores que serán admitidos por los atributos, es decir, su tamaño y formato. Es importante que este análisis se lo lleve a cabo de manera minuciosa porque si se llegase a escoger un tamaño o un tipo de dato que no satisfaga las reglas del negocio inminentemente ocasionaría alteraciones en el buen desempeño de la aplicación.

iv) Determinar los atributos de las claves candidatas, primarias y alternativas

Haciendo uso de las buenas prácticas de desarrollo con el marco de trabajo Django con PostgreSQL, se fijan las claves primarias de la siguiente manera:

- Usuario: id_user (Incluye tanto a Administradores como clientes)
- Mesa: id_table
- Pedido: id_order
- Categoría: id_category
- Alimento: id_food

La representación de las fases i, ii, iii y iv del diseño conceptual de la base de datos se puede observar en la sección correspondiente a Diccionario de datos en cada una de las tablas correspondientes a las entidades del sistema.

b) Diseño lógico

En esta etapa del diseño los datos se dotan de una estructura además de establecer relaciones entre ellos. En este sentido, el modelo lógico añade más información al modelo conceptual.

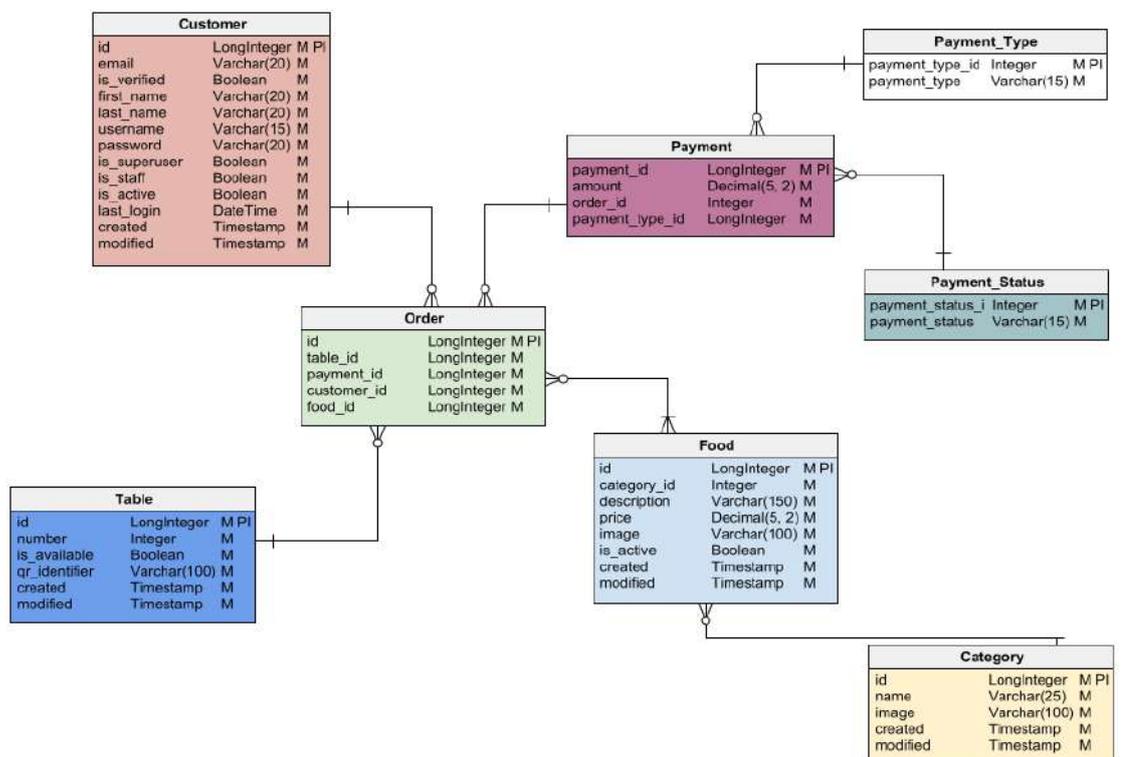


Ilustración 7-3: Diseño y modelado de datos lógico

Realizado por: Cuenca A.; Llinin F., 2022.

Tabla 5-3: Diccionario de datos de la tabla User

Nombre de la tabla	User						
Fuente de la tabla	Los datos de la tabla 'User' serán obtenidos por cada uno de los usuarios administradores, empleados y clientes que se vayan registrando en la aplicación informática.						
Descripción	Esta tabla almacena los datos correspondientes a los usuarios de la aplicación informática iCard.						
Documentos actualizados	23/05/2022						
Nombre del campo	Tipo de dato	Atributos	PK	FK	Bytes	Valor por defecto	Descripción
id	bigint	No nulo	X	-	64	-	Clave primaria para la tabla 'User'
password	character varying	No nulo	-	-	20	-	Contraseña del usuario.
last_login	timestamp with time zone	Nulo	-	-	6	-	Fecha del último registro de ingreso a la aplicación.
is_superuser	boolean	No nulo	-	-	1	false	Si el usuario es un usuario administrador.
created	timestamp with time zone	No nulo	-	-	6	-	Fecha de creación del registro.
modified	timestamp with time zone	No nulo	-	-	6	-	Fecha de actualización del registro.
username	character varying	No nulo	-	-	15	-	Nombre de usuario.
email	character varying	No nulo	-	-	30	-	Correo electrónico del usuario.
first_name	character varying	No nulo	-	-	25	-	Nombres del usuario
last_name	character varying	No nulo	-	-	25	-	Apellidos del usuario
is_verified	boolean	No nulo	-	-	1	false	Si el usuario está verificado por medio de su correo electrónico.
is_staff	boolean	No nulo	-	-	1	false	Si el usuario pertenece al personal del restaurante.
is_active	boolean	No nulo	-	-	1	true	Si el usuario está activo a nivel de aplicación.

Realizado por: Cuenca A.; Llinin F., 2022.

3.3.4.3. Interfaces de usuario

Con la ayuda de las reuniones realizadas con el administrador de la pizzería “D’Baggio” se definieron el estándar de las interfaces de usuario que se desarrollarán en la aplicación web tanto para los administradores como para los clientes. En este sentido, se han considerado los colores de esta, el tipo de letra, la iconografía y el posicionamiento del logo de la aplicación. Todos estos aspectos con el afán de que la interfaz que observará el usuario final resulte sencilla y amigable.

a) Story board

El flujo del storyboard se cuenta desde que el consumidor accede a una mesa habiendo escaneado el código QR hasta que pide la cuenta de su orden. Una vez se ha completado el proceso de licitación de requisitos entonces se puede proceder con el storyboard que tiene por objetivo servir como un marco básico y de guía para el diseño de las interfaces de usuario del sitio web My iCard.

b) Modelo conceptual

El propósito del modelo conceptual es ayudar a realizar el diseño de la aplicación web en forma de prototipo de papel, wireframe o diseño. Así en la **Tabla 6-3** se presenta el modelo conceptual que se utilizará en la construcción de la aplicación web.

Tabla 6-3: Métodos, Técnicas y Fuentes para el Trabajo de Integración Curricular

Número	Tarea	Elemento	Información
1	Página de inicio (Landing page)	Cabecera (Header)	Contiene espacio para logotipos de imágenes, botón de inicio de sesión, botón de registro, en lace a la página de información, precios.
		Pie de página (Footer)	Contiene identidad corporativa (dirección, teléfono, correo electrónico), enlace a la página de información, medios de comunicación social, formas de pago, derechos de autor.
		Contenido	Contiene imágenes hero, cómo funciona la aplicación, beneficios de la aplicación.
2	Tablero de mandos de pedidos	Cabecera (Header)	Contiene espacio para logotipos de imágenes, botón de inicio de sesión, botón de registro, en lace a la página de información, precios.
		Título de sección	Texto principal que muestra el nombre de la vista actual “Pedidos”

		Botones	Botón para recargar mesas de forma manual y automática.
		Estado de mesas	Detalle gráfico de la mesa, con sus respectivas órdenes y estados por color para indicar si hay pedidos, solicitud de cuenta o si se ha cerrado.
3	Tablero de mandos de mesas	Cabecera (Header)	Contiene el logo de la aplicación, saludo al usuario que ha iniciado sesión y el botón de salida de sesión.
		Título de sección	Texto principal que muestra el nombre de la vista actual “Mesas”.
		Botones	Botón para crear una nueva mesa.
		Detalles de la mesa	Rejilla de datos de las mesas (Número de mesa, Disponibilidad de mesa y acciones de editar y eliminar).
4	Tablero de mandos de pagos	Cabecera (Header)	Contiene el logo de la aplicación, saludo al usuario que ha iniciado sesión y el botón de salida de sesión.
		Título de sección	Texto principal que muestra el nombre de la vista actual “Pagos”.
		Botones	Botón para establecer el pago de una mesa.
5	Tablero de mandos de categorías	Cabecera (Header)	Contiene el logo de la aplicación, saludo al usuario que ha iniciado sesión y el botón de salida de sesión.
		Título de sección	Texto principal que muestra el nombre de la vista actual “Categorías”.
		Botones	Botón para crear una nueva categoría.
		Detalles de la categoría	Rejilla de datos de las categorías (Imagen, Nombre y acciones de editar y eliminar).
6	Tablero de mandos de alimentos	Cabecera (Header)	Contiene el logo de la aplicación, saludo al usuario que ha iniciado sesión y el botón de salida de sesión.
		Título de sección	Texto principal que muestra el nombre de la vista actual “Alimentos”.
		Botones	Botón para crear un nuevo alimento.
		Detalles del alimento	Rejilla de datos de los alimentos (Imagen, Nombre, Precio, Categoría, Descripción, Disponibilidad y acciones de editar y eliminar).
7	Tablero de mandos de usuarios	Cabecera (Header)	Contiene el logo de la aplicación, saludo al usuario que ha iniciado sesión y el botón de salida de sesión.
		Título de sección	Texto principal que muestra el nombre de la vista actual “Usuarios”.
		Botones	Botón para crear un nuevo alimento.
		Detalles del alimento	Rejilla de datos de los usuarios (Usuario, Correo electrónico, Nombres, Apellidos, Si está activo, Si es administrador y acciones de editar y eliminar).

Realizado por: Cuenca A.; Llinin F., 2022.

c) Wireframes

El objetivo del diseño wireframe es facilitar la creación de las interfaces en la etapa de desarrollo de estas. La maquetación se basa exclusivamente en el modelo conceptual

desarrollado con anterioridad y partiendo desde la cabecera (parte superior de la página) hasta el pie de página (parte inferior de la página). A continuación, se muestran los wireframes de los cuales estará compuesta la aplicación My iCard:

El diseño del wireframe de la página de inicio consta de una cabecera que contiene el logo de la aplicación, el botón de recursos, ayuda, precios e iniciar sesión. A continuación, se compone del texto e imagen hero así como de información básica acerca de la aplicación. El wireframe de la página de inicio se puede ver en la **Ilustración 9-3**. El diseño wireframe de la página de inicio de sesión de la aplicación web “My iCard” consiste en una página con un fondo compuesto por un fondo del color primario de la aplicación. En el centro un cuadro que contiene los cuadros de entrada de texto y el botón para iniciar sesión. El wireframe de la página de inicio de sesión se puede ver en la **Ilustración 10-3**.

d) Diseño como escenario

Esta etapa del diseño de interfaces tiene como objetivo determinar el proceso de diseño del sistema y los datos requeridos por el sistema para crear y diseñar los mockups de los wireframes que se han hecho (Paramarini, 2019) . Por tanto, después de obtener retroalimentación de las entrevistas con el administrador y de las historias de usuarios, se puede establecer el detalle gráfico en forma visual. Este diseño consta de un logotipo, la paleta de colores que se utilizará en el sitio web y el tipo de letra que se utilizará en el sitio web.

Dentro de la aplicación web “My iCard” el logotipo se utilizará como un tipo de identidad principal para remarcar la aplicación. El logotipo se puede hallar en la parte izquierda superior de las páginas del control de mandos para que pueda ser reconocido por los consumidores. El logotipo cuenta con dos modelos, el primero con los colores estándar de la aplicación y el segundo con el fondo oscuro para lugares como el pie de página de la página principal. Los logotipos descritos anteriormente se pueden observar en la **Ilustración 11-3**. La selección de la paleta de colores principales en las páginas de la aplicación “My iCard” se realiza en base a los colores de los logotipos. Hay dos colores principales, el color “Marigold” de código #F2A71E y el color “Violeta oscuro” de código #332B3D. Los colores para los botones dependerán de las acciones que se vayan a realizar al pulsarlo. Como para el desarrollo de la aplicación se usa Semantic UI, se usan los colores para botones primarios y de cancelación. Para las demás acciones, se usa el color “Naranja Princeton” de código #F2791D. La paleta de colores se puede ver en la **Ilustración 12-3**.

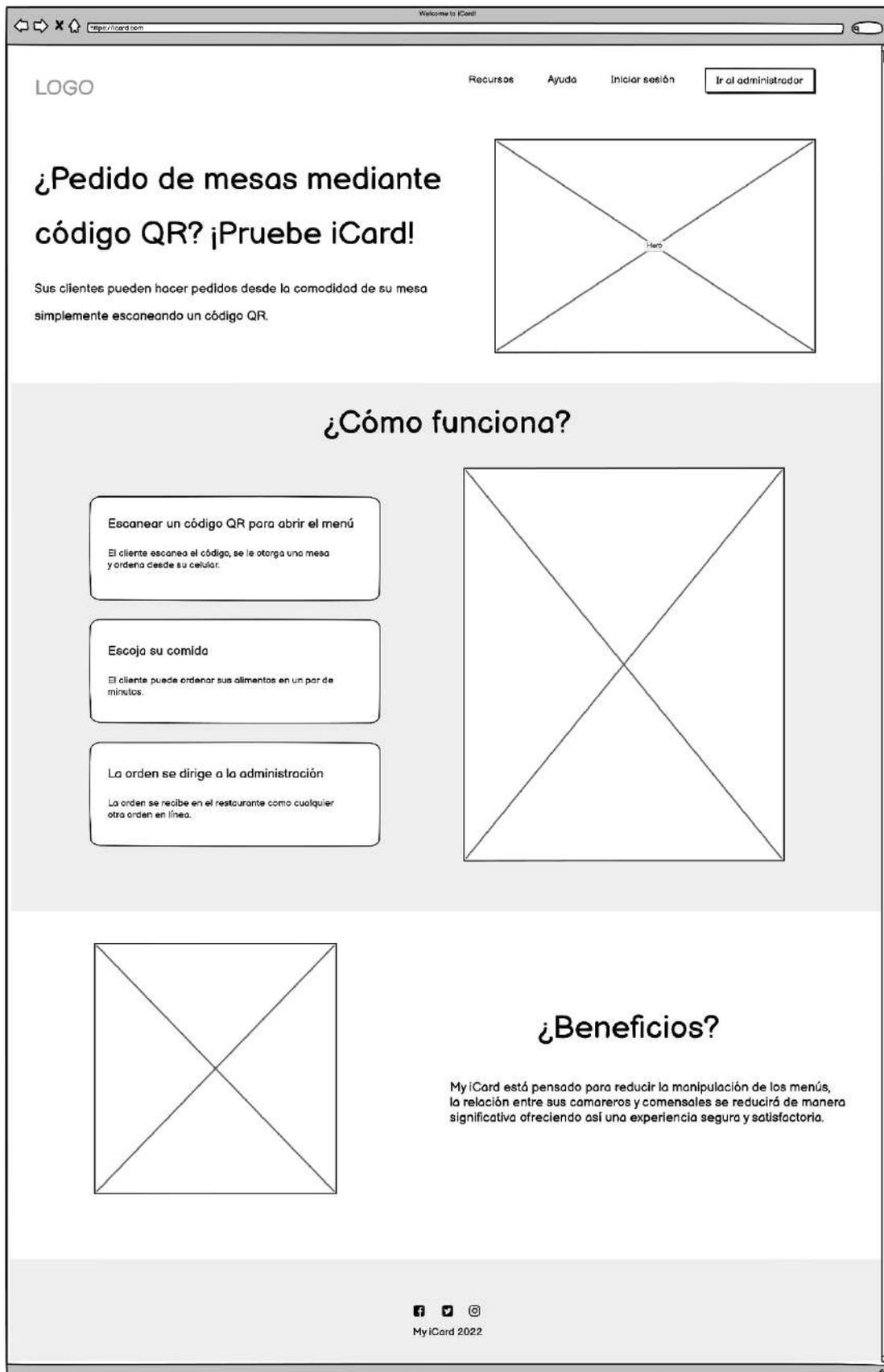


Ilustración 9-3: Diseño del wireframe de la página de inicio

Realizado por: Cuenca A.; Llinin F., 2022.

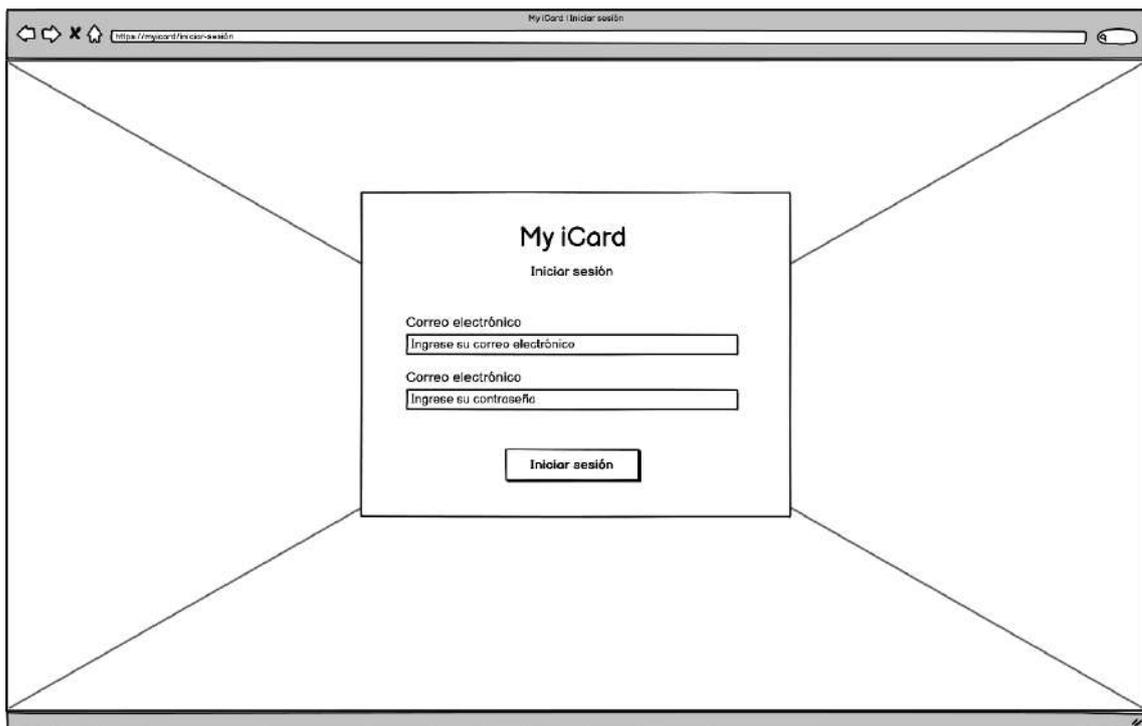


Ilustración 10-3: Diseño del wireframe del inicio de sesión.

Realizado por: Cuenca A.; Llinin F., 2022

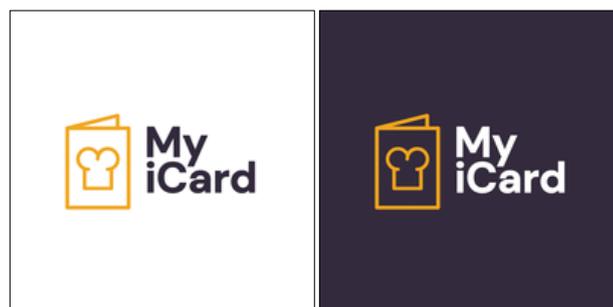


Ilustración 11-3: Logos de “My iCard”

Realizado por: Cuenca A.; Llinin F., 2022.

En lo que respecta a tipografía, “My iCard” desea mostrar comodidad y suavidad tanto para los administradores y clientes. De esta manera, el tipo de letra escogido es Lato que es un tipo de estructura Sans-Serif. El espesor de la fuente variará entre “Regular” y “Bold” dependiendo la acentuación del contenido. La fuente escogida se usará en todas las páginas que componen la aplicación.

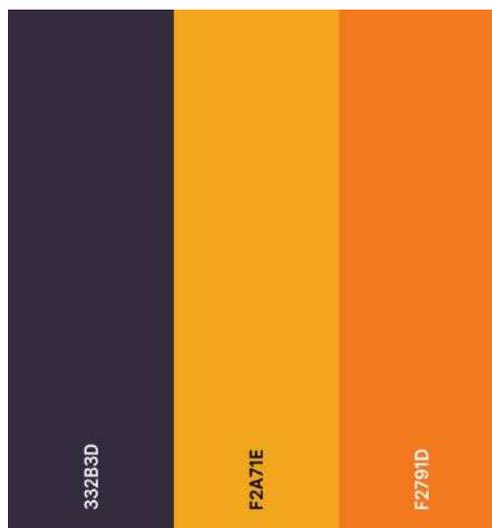


Ilustración 12-3: Selección de paleta de colores primarios para la aplicación “My iCard”

Realizado por: Cuenca A.; Llinin F., 2022.

3.3.5. *Desarrollo*

3.3.5.1. *Estándar de codificación*

El estándar de codificación se define para que la codificación de la aplicación se realice de forma ordenada, mantenerla limpia y consecuentemente bien documentada. Esto que la aplicación mantenga su calidad y que en posteriores fases sea fácil de mantener y actualizar.

a) Estándar de codificación de la base de datos

Tabla 7-3: Estándar de codificación para la base de datos

Lenguaje	SQL	
Versión	SQL 3.7.2	
Estándar	PostgreSQL C99	
Aspecto	Características	Descripción
Tablas	Nombres	<ul style="list-style-type: none"> Nombres colectivos o en forma plural. Ningún nombre de tabla puede llevar el mismo nombre de una columna Sin concatenaciones por guion o sub-guión.
	Prefijos	<ul style="list-style-type: none"> No existe uso de ningún prefijo antes de cada nombre de tabla.
Columnas	Nombre	<ul style="list-style-type: none"> Nombres en singular En mayúsculas, salvo el caso de un nombre propio.

	Identificadores	<ul style="list-style-type: none"> Los identificadores principales siempre serán “id”
Convención de nomenclatura	Clases	<ul style="list-style-type: none"> Unido a un guion bajo y en minúsculas siguiendo la nomenclatura “snake_case”
	Columnas	<ul style="list-style-type: none"> Unido a un guion bajo y en minúsculas siguiendo la nomenclatura “snake_case”

Realizado por: Cuenca A.; Llinin F., 2022.

b) Estándar de codificación del back-end

Tabla 8-3: Estándar de codificación para el back-end

Lenguaje	Python	
Versión	Python 3.7.2	
Estándar	PEP8	
Aspecto	Características	Descripción
Disposición del código	Importaciones	<ul style="list-style-type: none"> Secuencia determinada. Primero las bibliotecas estándar, luego las de terceros y finalmente las bibliotecas locales. Si se necesita una única función entonces se realiza una importación absoluta.
	Espacios en blanco	<ul style="list-style-type: none"> Dos espacios en blanco deben rodear las clases y las funciones. Un espacio en blanco en cada método de una clase. Un espacio en blanco entre cada bloque de importaciones.
	Indentaciones	<ul style="list-style-type: none"> Indentación por tabulación.
Longitud y saltos de línea	Longitud	<ul style="list-style-type: none"> Inferior a 79 caracteres.
	Saltos de línea	<ul style="list-style-type: none"> En sentencias con operadores binarios se incluye una nueva línea.
Espacios y comillas de cadena	Espacios	<ul style="list-style-type: none"> Un solo espacio en blanco alrededor de ambos lados de un operador, uno después de la coma y ninguno dentro de la apertura o cierre de paréntesis.
	Comillas	<ul style="list-style-type: none"> Comillas dobles
Convención de nomenclatura	Variables	<ul style="list-style-type: none"> Nombres gramaticalmente correctos. En caso de palabras que lleven la letra ñ, se usará la notación “ni”
	Clases	<ul style="list-style-type: none"> Empieza con mayúscula y sigue la nomenclatura “camelCase” en caso de usar más de dos palabras.
	Funciones	<ul style="list-style-type: none"> Unido a un guion bajo y en minúsculas siguiendo la nomenclatura “snake_case”
	Argumentos de funciones	<ul style="list-style-type: none"> El “self” siempre va como primer argumento para declarar una variable de instancia.
	Constantes	<ul style="list-style-type: none"> Se declaran en mayúsculas
Documentación	Métodos (Funciones)	<ul style="list-style-type: none"> Se especifica por cada método crítico los parámetros, el tipo de retorno y los tipos de datos.

Realizado por: Cuenca A.; Llinin F., 2022.

c) Estándar de codificación del Front-end

Tabla 9-3: Estándar de codificación para el Front-end

Lenguaje	Javascript	
Versión	Javascript 1.7	
Estándar	ES6	
Aspecto	Características	Descripción
Disposición del código	Importaciones	<ul style="list-style-type: none"> Todas las importaciones irán en la parte superior de cada archivo. Secuencia determinada. Primero las propias de React, librerías de terceros, librerías anónimas y finalmente los archivos. Todas las importaciones estarán en líneas distintas.
	Archivo index.js	<ul style="list-style-type: none"> Cada componente contendrá su archivo "index.js" que se encargará de exportar todo el contenido de este.
	Clases para estilos CSS	<ul style="list-style-type: none"> Cada componente contendrá su archivo SCSS que se encargará de recoger todas las clases del componente en un único archivo.
	Indentaciones	<ul style="list-style-type: none"> Indentación por tabulación.
Longitud y saltos de línea	Longitud	<ul style="list-style-type: none"> Inferior a 79 caracteres.
	Saltos de línea	<ul style="list-style-type: none"> En sentencias con operadores binarios se incluye una nueva línea.
Espacios y comillas de cadena	Espacios	<ul style="list-style-type: none"> Un solo espacio en blanco alrededor de ambos lados de un operador, uno después de la coma y ninguno dentro de la apertura o cierre de paréntesis.
	Comillas	<ul style="list-style-type: none"> Comillas dobles para los atributos JSX y comillas simples para el código JS.
Convención de nomenclatura	Archivos	<ul style="list-style-type: none"> Se usa "PascalCase" con la extensión .js
	Componente	<ul style="list-style-type: none"> Nombre igual que el componente
	Variables	<ul style="list-style-type: none"> Nombres gramaticalmente correctos. En caso de palabras que lleven la letra ñ, se usará la notación "ni"
	Clases	<ul style="list-style-type: none"> Similar a los componentes. Se sigue la nomenclatura "PascalCase"
	Funciones	<ul style="list-style-type: none"> Se sigue la nomenclatura "camelCase"
	Constantes	<ul style="list-style-type: none"> Se declaran en mayúsculas
Archivos y carpetas	Estructura de directorios	<p>La aplicación se estructura de la siguiente manera:</p> <ul style="list-style-type: none"> src <ul style="list-style-type: none"> assets: Contiene activos estáticos como imágenes, svgs, logotipo de la empresa, etc. components: componentes reutilizables como barra de navegación, botones, formularios services: módulos de JavaScript.

		<ul style="list-style-type: none"> ○ hooks: métodos redux ○ utils: utilidades, ayudantes, constantes. ○ views/pages: páginas de la aplicación ○ index.js ○ App.js
	Carpeta de componente	<ul style="list-style-type: none"> • Contendrá sus respectivos archivos agrupados en esta. Los archivos necesarios son: Componente.js, Componente.scss y Componente.test.js (En caso de hacer test del componente)
Documentación	Métodos (Funciones)	<ul style="list-style-type: none"> • Se especifica por cada método crítico los parámetros, el tipo de retorno y los tipos de datos.

Realizado por: Cuenca A.; Llinin F., 2022.

3.3.5.2. Configuración de los entornos de desarrollo

Para poder iniciar con el desarrollo del producto, antes se deben realizar varias configuraciones con respecto a las herramientas que serán usadas para crear el sistema. Aunque en la actualidad existen un sinfín de maneras de poder configurar herramientas de desarrollo, a continuación, se describe la configuración que ha funcionado para este proyecto específico. Para poder configurar un entorno similar al que se va a usar en este proyecto se debe contar con la preinstalación de:

- NodeJS 16.13.1 con yarn 1.22.15
- Python 3.7.8 con pip

Las instalaciones extras que se realicen a lo largo del desarrollo del sistema se irán detallando a medida que sean necesarias.

a) Entorno Front-End

Para empezar a trabajar con React se ha utilizado la línea de comandos create-react-app. La creación o instalación de un nuevo proyecto de ReactJS cuenta con tres pasos y el primero de ellos es crear una nueva aplicación ejecutando el comando siguiente:

```
$ npx create-react-app
```

Ilustración 13-3: Comando para la creación de un nuevo proyecto en ReactJS

Realizado por: Cuenca A.; Llinin F., 2022.

Para evitar que la utilidad del comando create-react-app se instale de manera global, se usa npx para instalar la utilidad localmente. De esta el desarrollador se asegura de que una vez se termine de usarla pueda ser desechada. La duración de la instalación dura un momento y da

como resultado una carpeta con el nombre que se le haya otorgado a la aplicación. El servidor de desarrollo o local puede ser activado moviéndose dentro de la carpeta de la aplicación y ejecutando el siguiente comando:

```
$ yarn start
```

Ilustración 14-3: Comando para iniciar servidor local en ReactJS

Realizado por: Cuenca A.; Llinin F., 2022.

Al ejecutar el comando, una ventana del navegador predeterminado abre una nueva ventana en la dirección `http://localhost:3000` o, `http://127.0.0.1` donde se muestra la plantilla del proyecto ReactJS. Para completar la configuración base, es necesario instalar Semantic UI React.

Semantic UI React proporciona una cantidad significativa de componentes React. Esto hace que, mediante la importación de componentes se evite crear los componentes desde cero y el desarrollador pueda centrar sus esfuerzos en la lógica de programación al componente. Para el caso de “My iCard”, los estilos de los componentes se trabajarán desde la hoja de estilos de componentes de Semantic UI, pero para algunos casos, se realizarán estilos personalizados mediante el uso de hojas de estilo SCSS. Ambas dependencias pueden ser instaladas con yarn como se muestra a continuación:

```
$ yarn add semantic-ui-react semantic-ui-css  
$ yarn add sass
```

Ilustración 15-3: Comando para añadidura de Semantic Ui React y Sass

Realizado por: Cuenca A.; Llinin F., 2022.

Para finalizar, se debe realizar una importación para el archivo CSS global en el archivo `src/index.js` para finalizar la instalación.

```
$ import 'semantic-ui-css/semantic.min.css'  
$ import './scss/global.scss'
```

Ilustración 16-3: Comando para la importación de Semantic UI React y Sass

Realizado por: Cuenca A.; Llinin F., 2022.

b) Entorno back-end

Debido a la arquitectura y el tipo de solución que se ofrece con el proyecto, la configuración de Django no será tan fácil como la de ReactJS. Aunque la documentación de DJANGO ofrece una forma sencilla de poder iniciar con un proyecto web, en este caso no se hará uso de las configuraciones básicas. Antes que nada, se debe crear un proyecto Django con ayuda del comando:

```
§ django-admin startproject
```

Ilustración 17-3: Comando para la creación de un nuevo proyecto en Django

Realizado por: Cuenca A.; Llinin F., 2022.

La duración de la instalación dura un momento y da como resultado una carpeta con el nombre que se le haya otorgado al proyecto.

Hay que destacar que, según las buenas prácticas marcadas por la documentación oficial de Django, se debe contar con la configuración de variables de entorno, así como de una distribución adecuada de archivos de configuración para facilitar el manejo de los entornos de desarrollo. En este sentido, la tabla presenta la configuración elegida para este proyecto.

Tabla 10-3: Estándar de codificación para la base de datos

Directorio raíz	Archivo	Descripción
settings/	base.py	Configuraciones básicas del proyecto. Serán usadas tanto por el archivo LOCAL.PY y PROD.PY. Ej.: Variables de entorno, directorio raíz, zona de tiempo, lenguaje, etc.
	local.py	Configuraciones del entorno de desarrollo en el servidor local (localhost). Esta configuración únicamente será usada antes del despliegue. Ej.: Base de datos del local, hosts permitidos, DEBUG, Middlewares, etc.
	prod.py	Configuraciones del entorno de desarrollo en el servidor en la nube (AWS). Esta configuración únicamente será en el despliegue. Ej.: RDS para base de datos en producción, hosts permitidos, DEBUG, Middlewares, etc.

Realizado por: Cuenca A.; Llinin F., 2022.

Ahora se continúa con la configuración de la base de datos. En este caso se hará uso de PostgreSQL por lo que se necesita una nueva configuración.

Antes que nada, es importante tener la CLI de SQL y haber creado una base de datos con sus respectivas credenciales (usuario y contraseña de administrador). Finalmente, dentro de settings/local.py del directorio raíz, se debe configurar la base de datos de la siguiente manera:

```

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql_psycopg2",
        "NAME": env("DATABASE_NAME"),
        "USER": env("DATABASE_USER"),
        "PASSWORD": env("DATABASE_PASSWORD"),
        "HOST": env("DATABASE_HOST"),
        "PORT": env("DATABASE_PORT"),
        "TEST": {
            "MIRROR": "default",
        },
    },
}

```

Ilustración 18-3: Configuración de conexión de una base de datos PostgreSQL en Django

Realizado por: Cuenca A.; Llinin F., 2022.

Si el sistema operativo utilizado para el desarrollo es Windows, es importante que el módulo “psycopg2” de PostgreSQL para Python sea añadido al archivo requirements/base.txt. Para culminar con el proceso de la configuración básica del entorno back-end, se debe crear un super usuario con:

```
$ python manage.py createsuperuser
```

Ilustración 19-3: Comando para crear un superusuario de un proyecto en Django

Realizado por: Cuenca A.; Llinin F., 2022.

Esto solicita un nombre de usuario y una contraseña para la cuenta del administrador principal del proyecto Django. Con estas configuraciones, concluye el proceso de instalación el entorno del back-end.

c) Proyecto Django

En esta sección se presenta, de forma parcial y muy resumida, el proyecto “My iCard” y se ve parte del código a manera de ejemplo de cómo se hacen los modelos de Django, se serializan y cómo luego se accede a esos datos desde una aplicación React. A lo largo de la sección se creará una aplicación de Django. A continuación, se crea su respectivo modelo y consecuentemente su serializador para al final, crear un proceso de consumo de los datos de esta API.

i) Aplicaciones de Django

Toda la funcionalidad de Django se basa en la modularidad de funciones, es decir, crear aplicaciones “pequeñas” para que las interacciones con las demás funcionalidades sean lo más reducidas posibles. Cada aplicación cuenta con su propia configuración de direcciones URL, modelos, serializadores, etc. La ubicación de la nueva aplicación depende de que tan organizado se encuentre el proyecto. En este caso, y siguiendo las buenas prácticas de Django, se crea un directorio único para las aplicaciones. Finalmente, se crea una aplicación de la siguiente manera:

```
$ django-admin startapp pagos
```

Ilustración 20-3: Comando para crear una aplicación en un proyecto en Django

Realizado por: Cuenca A.; Llinin F., 2022.

El comando anterior crear una nueva aplicación llamada “payments” con una estructura de carpetas que incluyen todos los ficheros mencionados en la introducción de esta sección y que son necesarios. Esto permite ahorrar tiempo en el desarrollo además de que permite trabajar con los archivos de una manera mucho más ordenada y limpia. Para Django pueda reconocer la nueva aplicación se la debe incluir en la configuración del proyecto. Dentro de settings/base.py se debe incluir dentro de las aplicaciones instaladas de Django. En este caso y siguiendo las buenas prácticas de desarrollo en Django, se han separado las aplicaciones en terceros, locales y aplicaciones de Django.

```
DJANGO_APPS = (  
    ...  
    "django.contrib.admin",  
    ...  
)  
  
LOCAL_APPS = (  
    ...  
    "applications.payments",  
    ...  
)  
  
THIRD_PARTY_APPS = (  
    ...  
    "rest_framework",  
    ...  
)  
  
INSTALLED_APPS = DJANGO_APPS + LOCAL_APPS + THIRD_PARTY_APPS
```

Ilustración 21-3: Configuración de aplicaciones en un proyecto en Django

Realizado por: Cuenca A.; Llinin F., 2022.

Para crear un modelo dentro de una aplicación se requiere especificar los campos que se necesiten. Con este fin, Django proporciona varias opciones para la creación de los modelos. Para el caso de la clase Payment por ejemplo se cuentan con los campos: “table”, “amount”, “payment_type”, “payment_status”. Así también, se definen propiedades como el valor máximo, las posiciones decimales, las elecciones, etc. A continuación, se puede observar el modelo:

```
from django.db import models
from model_utils.models import TimeStampedModel

from applications.tables.models import Table

PAYMENT_TYPES_ENUM = (("0", "EFECTIVO"), ("1", "TARJETA DE
CRÉDITO/DEBITO"))
PAYMENT_STATUS_ENUM = (("0", "PENDIENTE"), ("1", "PAGADO"))

class Payment(TimeStampedModel):
    table = models.ForeignKey(Table, on_delete=models.SET_NULL,
null=True)
    amount = models.DecimalField("Total", max_digits=10,
decimal_places=2)
    payment_type = models.CharField(
        "Tipo de pago", max_length=20, choices=PAYMENT_TYPES_ENUM,
default=0
    )
    payment_status = models.CharField(
        "Estado del pago", max_length=15, choices=PAYMENT_STATUS_ENUM,
default=0
    )
```

Ilustración 22-3: Clase Payment dentro de payments/models.py

Realizado por: Cuenca A.; Llinin F., 2022.

Debido al alcance de este proyecto, las marcas de tiempo son un requisito fundamental por lo cual se debe anotar que la clase Payment toma una clase abstracta TimeStampedModel que añade de forma automática los campos “created” y “modified”. De esta forma, cada vez que se cree o actualice un registro de la base de datos su marca de tiempo será registrada sin necesidad de hacer algún proceso extra. En caso de contar restricciones en los datos y/o relaciones entre tablas Django ofrece la posibilidad de validar y procesar estas relaciones, por ejemplo, la siguiente línea indica que el campo “amount” será del tipo decimal, visualmente se observará como “Total”, el total de dígitos que tendrá es de diez y los decimales que contendrá son dos.

```
amount = models.DecimalField("Total", max_digits=10, decimal_places=2
```

Ilustración 23-3: Campo “amount” de payments/models.py con una propiedad de validación

Realizado por: Cuenca A.; Llinin F., 2022.

Asimismo, es recomendado hacer un sobrecargo sobre la función `__str__` en cada modelo, esto con el fin de que sea mucho más fácil identificar los objetos de una tabla. Además, para asegurar la UX de usuario, se debe asegurar de que los términos plurales y singulares concuerden con las reglas gramaticales del idioma en el que se usará la aplicación por lo que de igual forma se debe definir la clase `Meta`.

```
class Meta:
    verbose_name = "Pago"
    verbose_name_plural = "Pagos"

def __str__(self):
    return str(self.table)
```

Ilustración 24-3: Sobrecarga del método `__str__` y definición de la clase `Meta`

Realizado por: Cuenca A.; Llinin F., 2022.

Los modelos que hayan sido creados y registrados en `admin.py` de la aplicación se podrán observar desde panel de administración de Django. Esto hace posible que los registros almacenados puedan ser modificados, eliminados, filtrados, etc. Algunos pagos creados del modelo `Payment` se pueden observar en la **Ilustración 25-3**.

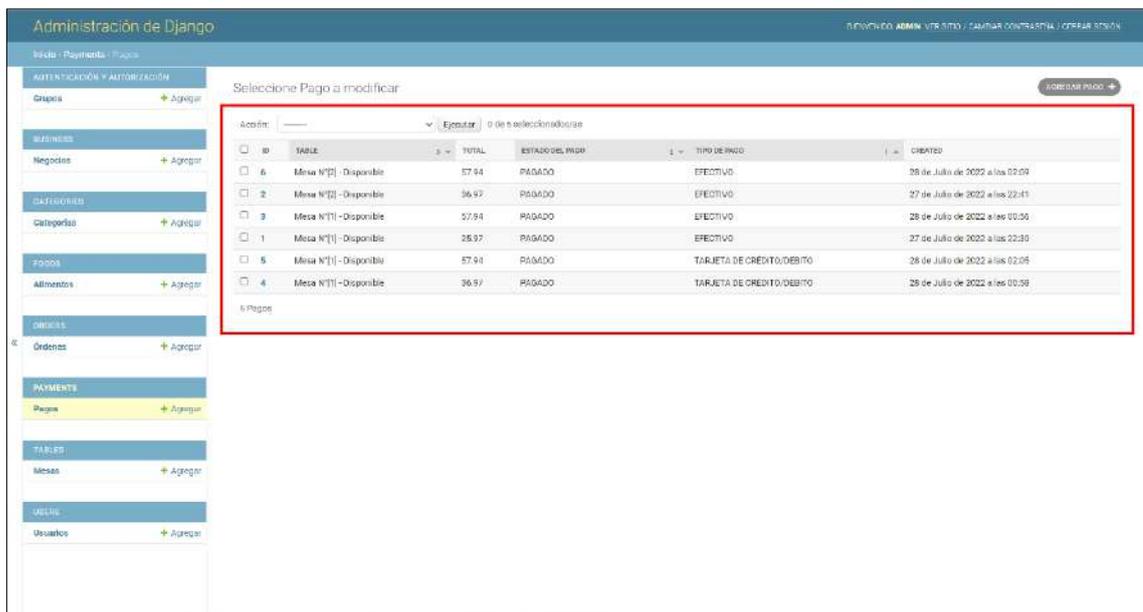


Ilustración 25-3: Aplicación Payment dentro del panel de administración de Django

Realizado por: Cuenca A.; Llinin F., 2022.

Cada aplicación de Django tiene su propio fichero de configuración para el administrador (admin.py) donde se puede personalizar cada una de las opciones que se podrán observar desde el administrador. Para la clase Payment la configuración es la que muestra la **Ilustración 26-3**.

```
from django.contrib import admin
from .models import Payment

@admin.register(Payment)
class PaymentAdmin(admin.ModelAdmin):
    list_display = [
        "id",
        "table",
        "amount",
        "payment_status",
        "payment_type",
        "created",
    ]
```

Ilustración 26-3: Configuraciones del panel de administración de payments/admin.py

Realizado por: Cuenca A.; Llinin F., 2022.

Django, con respecto a la migración de datos permite ir actualizando la base de datos a medida que el desarrollo va avanzando. Con el concepto de migraciones el desarrollador no tiene la necesidad de reiniciar la base de datos cada vez que ocurre algún cambio dentro de esta. Para que esto sea posible, Django utiliza dos pasos para las migraciones. La primera parte se encarga de analizar los posibles conflictos que puedan surgir con respecto a las validaciones de los campos.

```
$ python manage.py makemigrations
```

Ilustración 27-3: Comando para ejecutar migraciones

Realizado por: Cuenca A.; Llinin F., 2022.

Y el segundo, permite ejecutar la migración a la base de datos.

```
$ python manage.py migrate
```

Ilustración 28-3: Comando para ejecutar migraciones

Realizado por: Cuenca A.; Llinin F., 2022.

ii) Serializadores de Django

Un serializador ayuda a convertir datos de gran complejidad nativos de Django en un formato mucho más entendible y generalmente aceptado por la comunidad de desarrolladores como viene siendo JavaScript Object Notation o comúnmente conocido como JSON. Al atravesar los datos mediante un serializador existe una conversión muy compleja de parte del controlador de datos. Esta conversión se la realiza desde dos perspectivas, cuando se extraen datos de la base de datos y cuando se desea ingresar nuevos datos en esta por lo que este procedimiento ahorra muchos recursos en materia de tiempo. Para hacer uso de los serializadores, se debe realizar una configuración en el fichero urls.py de la aplicación. Asimismo, dependiendo de la complejidad de la aplicación, se puede o no configurar el fichero routers.py. Este último, depende enteramente de las necesidades y requerimientos de la aplicación que este siendo desarrollada.

```
from rest_framework.routers import DefaultRouter
from .viewsets import *

router = DefaultRouter()
router.register(r"pagos", PaymentAPIViewSet, basename="pagos")
urlpatterns = router.urls
```

Ilustración 29-3: Fichero payments/routers.py

Realizado por: Cuenca A.; Llinin F., 2022.

Cada conjunto de vistas debe tener su propio serializador dentro del directorio de la aplicación. Con el uso de los serializadores también se puede agregar validaciones de la información de manera que se pueden establecer filtros de los datos del serializador para asegurar la validez de la información.

```
from rest_framework import serializers
from .models import Payment
from applications.tables.serializers import TableSerializer

class PaymentSerializer(serializers.ModelSerializer):
    table_data = TableSerializer(source="table", read_only=True)

    class Meta:
        model = Payment
        fields = [
            "id",
            "table",
            "table_data",
            "amount",
            "payment_status",
            "payment_type",
            "created",
        ]
```

Ilustración 30-3: Serializador basado en el modelo de la clase Payment

Realizado por: Cuenca A.; Llinin F., 2022.

iii) Conexión de ReactJS a la API

Para esta sección, se demostrará el listado de todos los pagos que se han hecho en el establecimiento. Naturalmente, estos pagos se encuentran almacenados en la respectiva base de datos y se hará uso exclusivo de la API RESTful resultado de la fase anteriormente explicada. Es recomendado crear una estructura de carpetas de manera que a lo largo del desarrollo se mantenga y que a la vez permita la escalabilidad y manejo a largo plazo del proyecto.

En este caso, se presenta el componente Payments que tiene por objetivo consumir los pagos registrados y renderizarlos en una tabla con cada uno de los detalles del pago. La **Ilustración 31-3** muestra el componente en cuestión.

```

...
export function TablePayments(props) {
  ...
  return (
    <>
      {size(payments) === 0 ? (
        <div className="table-payments-no-content">
          <NoContentSVG
            style={{
              width: "500px",
              height: "400px",
              textAlign: "center",
            }}
          />
          <span>Sin contenido</span>
        </div>
      ) : (
        <Table className="table-payments-admin">
          <Table.Header>
            <Table.Row>
              <Table.HeaderCell>ID</Table.HeaderCell>
              <Table.HeaderCell>Mesa</Table.HeaderCell>
              <Table.HeaderCell>Total</Table.HeaderCell>
              <Table.HeaderCell>Tipo de pago</Table.HeaderCell>
              <Table.HeaderCell>Fecha</Table.HeaderCell>
              <Table.HeaderCell></Table.HeaderCell>
            </Table.Row>
          </Table.Header>

          <Table.Body>
            {map(payments, (payment, index) => (
              <Table.Row key={index}>
                <Table.Cell>{payment.id}</Table.Cell>
                <Table.Cell>{payment.table_data.number}</Table.Cell>
                <Table.Cell>${ payment.amount}</Table.Cell>
                <Table.Cell>
                  <Icon name={getIconPaymentName(payment.payment_type)} />
                </Table.Cell>
                <Table.Cell>
                  {moment(payment.created).format("DD/MM/YYYY - HH:MM:SS")}
                </Table.Cell>
                <Table.Cell text-align="right">
                  <Button icon onClick={() => showPaymentDetails(payment)}>
                    <Icon name="eye" />
                  </Button>
                </Table.Cell>
              </Table.Row>
            )
          )
        </Table.Body>
      </Table>
    )}
    <ModalBasic
      show={showModal}
      onClose={openCloseModal}
      title={titleModal}
      children={contentModal}
    />
  </>
);
}

```

Ilustración 31-3: Componente TablePayments

Realizado por: Cuenca A.; Llinin F., 2022.

Con respecto al consumo de los datos se ha optado por usar peticiones asíncronas de manera que se ha creado un directorio únicamente dedicado a realizar estas peticiones al servidor. De esta manera, el consumo dedicado los pagos se puede observar en la **Ilustración 32-3**.

```
...
export async function createPaymentApi(paymentData) {
  try {
    const url = `${BASE_API}/pagos/`;
    const params = {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(paymentData),
    };

    const response = await fetch(url, params);
    const result = await response.json();
    return result;
  } catch (error) {
    throw error;
  }
}

export async function getPaymentByTableApi(idTable) {
  try {
    const tableFilter = `table=${idTable}`;
    const statusFilter = `payment_status=${PAYMENT_STATUS.PENDING}`;
    const url = `${BASE_API}/pagos/?${tableFilter}&${statusFilter}`;
    const params = {
      headers: {
        "Content-Type": "application/json",
      },
    };

    const response = await fetch(url, params);
    const result = await response.json();
    return result;
  } catch (error) {
    throw error;
  }
}
...
```

Ilustración 32-3: Fichero con los métodos de consumo para los pagos

Realizado por: Cuenca A.; Llinin F., 2022

El componente luego de realizar la correspondiente petición de los datos al servidor, haber montado de los datos y procesado eso en el componte de Semantic UI muestra una tabla como lo muestra la **Ilustración 33-3**.

ID	Mesa	Total	Tipo de pago	Fecha
1	1	\$ 25,97	☑	27/07/2022 - 22:07:41
2	2	\$ 36,97	☑	27/07/2022 - 22:07:01
3	1	\$ 57,94	☑	28/07/2022 - 00:07:54
4	1	\$ 36,97	☑	28/07/2022 - 00:07:02
5	1	\$ 57,94	☑	28/07/2022 - 02:07:09
6	2	\$ 57,94	☑	28/07/2022 - 02:07:19

Ilustración 33-3: Vista de la lista que muestra los pagos realizados en el establecimiento

Realizado por: Cuenca A.; Llinin F., 2022.

3.3.6. Pruebas

Con la finalidad de asegurar la calidad de “My iCard” y buscando satisfacer las necesidades específicas del cliente se han establecido 28 pruebas de aceptación exitosas. Para el caso del producto de este proyecto se ha utilizado el enfoque de pruebas de aceptación basados en reglas. Las reglas se extraen de cada una de las historias de usuario descritas en etapas anteriores siguiendo el modelo como se observa en la **Tabla 11-3**.

Tabla 11-3: Indicadores para medir la eficiencia Formato de prueba de aceptación.

Prueba de Aceptación	
Código: PA-01	Usuario: Administrador
N° Historia Usuario que prueba: HU-01	
Nombre de historia: Acceso al sistema principal	
Programador responsable: Alex Cuenca - Francisco Llinin	
Descripción:	Los usuarios administradores podrán ingresar a los módulos del sistema comprobando su usuario y clave de acceso
Pasos de ejecución:	1. Ingresar al login del sistema 2. Ingresar el usuario y contraseña 3. Dar clic en Ingresar
Resultado esperado:	Si el usuario existe se ingresa al sistema, caso contrario se muestra un mensaje de alerta
Resultado obtenido:	Se obtiene el resultado esperado
Evaluación:	Aprobado

Realizado por: Cuenca A.; Llinin F., 2022.

3.3.7. *Despliegue*

Puesto que todo el proyecto gira alrededor de una solución SaaS, es indiscutible la necesidad de que el aplicativo se halle alojado en la nube. Es así como, luego de haber realizado un análisis teórico de las herramientas cloud que ofrece el mercado actual, se ha escogido a Amazon Web Services como el proveedor de servicio en la nube que alojará a “My iCard”. Actualmente, AWS ofrece más cien distintos tipos de servicios. Cada uno de los servicios se halla destinado para atender y ofrecer soluciones sobre áreas específicas. Aunque muchos de estos servicios pueden integrarse y funcionar en conjunto, casi siempre, resulta complejo escoger las herramientas específicas que la aplicación requiere para asegurar su funcionamiento (AWS 2019).

Debido al alcance y la definición que tiene el aplicativo resultado de este proyecto, se ha escogido el servicio de Elastic Beanstalk. Este servicio de AWS permite implementar y administrar aplicaciones de manera rápida en la nube. Asimismo, evita que el desarrollador se preocupe aspectos como la arquitectura, la complejidad de la administración de los entornos, la capacidad de aplicación, balanceo de carga, escalado, monitorización y un amplio etcétera. Adicional a esto, EBS, es compatible con una amplia gama de lenguajes y lógicamente esto incluye Python con Django. El objetivo de esta sección es presentar muy resumidamente el proceso realizado para desplegar la aplicación “My iCard”. Esto incluirá las configuraciones más sensibles que se hayan realizado, así como el detalle de herramientas extras que hayan hecho falta a medida que el despliegue va tomando forma.

3.3.7.1. *Back-end*

Para empezar con el despliegue es necesario contar con una cuenta en la página de Amazon Web Services, Al momento de crear una nueva cuenta en AWS, puede ser elegible para una cuenta de nivel gratuito de AWS.

a) Interfaz de línea de comandos de Elastic Beanstalk

La EB CLI es necesaria para poder desempeñar varias operaciones para poder desplegar y administrar las aplicaciones y ambientes de Elastic Beanstalk. Para este fin, se ha usado el método de instalación con pip. Para este propósito, se puede usar el comando de la **Ilustración 34-3**.

```
$ pip install awsebcli
```

Ilustración 34-3: Interfaz de línea de comandos para AWS EB

Realizado por: Cuenca A.; Llinin F., 2022.

Una vez se haya completado la instalación, se puede comprobar la versión instalada. Y en cas

i) Init

Dentro de la raíz del proyecto “My iCard”, se debe ejecutar el comando siguiente:

```
$ eb init
```

Ilustración 35-3: Inicializar un repositorio en AWS EB

Realizado por: Cuenca A.; Llinin F., 2022.

Luego de inicializar un repositorio en AWS EB, aparecen varias preguntas que se presentan a continuación:

- **Región por defecto:** Determina la región del ambiente de Elastic Beanstalk. Se deberá escoger la región que más cerca esté de los clientes que usarán la aplicación.
- **Nombre de la aplicación:** Es el nombre de la aplicación de Elastic Beanstalk. Es recomendado que se presione enter y dejar el nombre por defecto.
- **Plataforma y rama de la plataforma:** EB CLI reconocerá automáticamente que se está usando Python. Por lo que ofrece un stack de herramientas preparadas. Es recomendado usar "Python 3.8 running on 64bit Amazon Linux 2".
- **CodeCommit:** Para el caso de “My iCard”, no se está subiendo desde un repositorio de GitHub, sino desde el computador local.
- **SSH:** Debido al alcance de este proyecto, es necesario conectarse a instancias EC2 posteriormente.
- **Keypair:** Debido a la necesidad de usar EC2, se necesita un par de llaves (keypair) para poder conectarse.

Finalmente, se podrá observar un nuevo directorio “.elasticbeanstalk” que contendrá los datos que se han especificado anteriormente. El fichero config.yml, debe contener algo similar a lo de la **Ilustración 36-3**.

```
branch-defaults:
  default:
    environment: my-icard-saas-env
    group_suffix: null
global:
  application_name: my-icard-saas
  branch: null
  default_ec2_keyname: aws-eb
  default_platform: Python 3.7
  default_region: us-west-2
  include_git_submodules: true
  instance_profile: null
  platform_name: null
  platform_version: null
  profile: eb-cli
  repository: null
  sc: null
  workspace_type: Application
```

Ilustración 36-3: Fichero config.yml dentro de .elasticbeanstalk

Realizado por: Cuenca A.; Llinin F., 2022.

ii) Create

Con ayuda de este comando se puede crear un ambiente en el cual se pueda desplegar la aplicación:

```
$ eb create
```

Ilustración 37-3: Inicializar un entorno en AWS EB

Realizado por: Cuenca A.; Llinin F., 2022.

- **Nombre del entorno:** Es el nombre del entorno de Elastic Beanstalk. Es recomendado que se presione enter y dejar el nombre por defecto.
- **Prefijo CNAME DNS:** Indica el enlace desde el cual la aplicación será accesible. Por defecto se accede desde %cname%.%region%.elasticbeanstalk.com
- **Balanceador de carga:** Es utilizado para distribuir el tráfico entre las instancias de su entorno.

iii) Status

Una vez se haya realizado el despliegue de la aplicación se puede revisar el estado de esta usando el comando de la **Ilustración 38-3**.

```
$ eb status

Environment details for: my-icard-saas-env
  Application name: my-icard-saas
  Region: us-west-2
  Deployed Version: app-220727_111615392930
  Environment ID: e-fi9pjgh7tp
  Platform: arn:aws:elasticbeanstalk:us-west-2::platform/Python 3.7
running on 64bit Amazon Linux 2/3.3.15
  Tier: WebServer-Standard-1.0
  CNAME: my-icard-saas-env.eba-kvwxcg4uf.us-west-2.elasticbeanstalk.com
  Updated: 2022-07-27 16:16:44.032000+00:00
  Status: Ready
  Health: Grey
```

Ilustración 38-3: Estado de un entorno y aplicación en AWS EB

Realizado por: Cuenca A.; Llinin F., 2022.

iv) Open

Con ayuda del comando de la **Ilustración 39-3** se podrá abrir una nueva ventana a la dirección de dominio CNAME.

```
$ eb open
```

Ilustración 39-3: Estado de un entorno y aplicación en AWS EB

Realizado por: Cuenca A.; Llinin F., 2022.

v) Console

Con ayuda del comando de la **Ilustración 40-3** se podrá abrir una nueva ventana de la consola de Elastic Beanstalk.

```
$ eb console
```

Ilustración 40-3: Estado de un entorno y aplicación en AWS EB

Realizado por: Cuenca A.; Llinin F., 2022.

En la **Ilustración 41-3** se puede observar todo el resumen del entorno sobre el cual se está ejecutando el entorno de la aplicación “My iCard”.

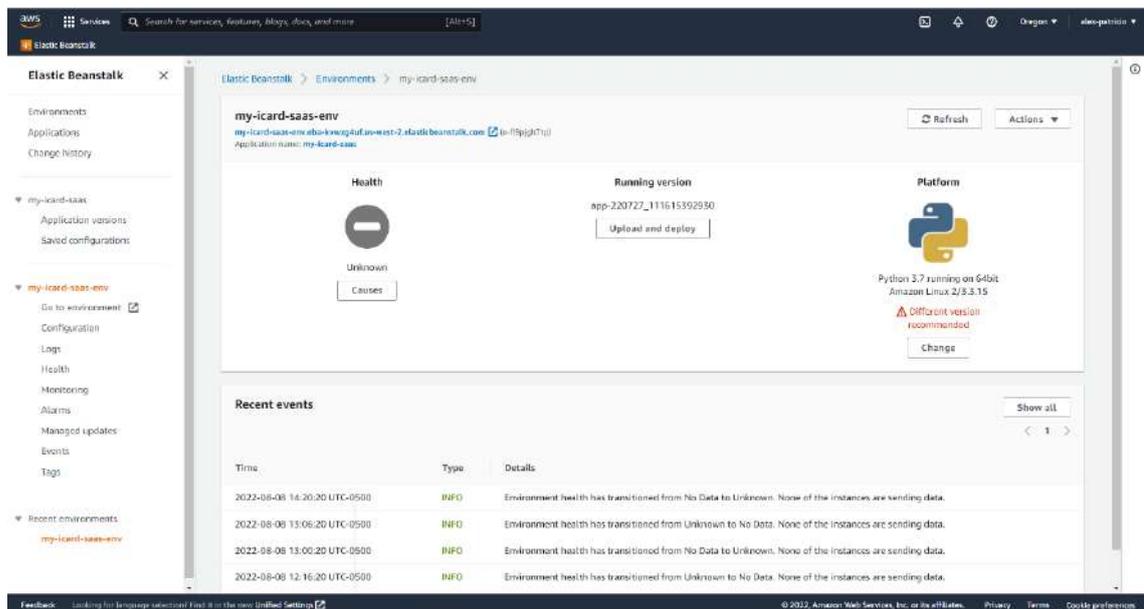


Ilustración 41-3: Consola del entorno sobre el cual se ejecuta “My iCard”

Realizado por: Cuenca A.; Llinin F., 2022.

b) Configuración de un entorno

Para tener un acceso correcto a la aplicación creada en la sección anterior se deben configurar:

- **PYTHONPATH:** Sirve para localizar los módulos de la aplicación.
- **application.py:** Elastic Beanstalk intenta lanzar la aplicación WSGI desde el fichero `application.py`.
- **DJANGO_SETTINGS_MODULE:** Es el fichero más importante que identifica cuales son las configuraciones que se van a usar.

Para empezar, se debe crear un nuevo directorio raíz con el nombre “.ebextensions”. Dentro de este, se debe definir lo descrito en la **Ilustración 42-3**.

```
option_settings:
  aws:elasticbeanstalk:container:python:
    WSGIPath: icard.wsgi:application
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /static: static_files
```

Ilustración 42-3: Fichero `django.config` dentro de “.ebextensions”

Realizado por: Cuenca A.; Llinin F., 2022.

Asimismo, antes de volver a desplegar la aplicación se debe añadir el CNAME en `ALLOWED_HOSTS` dentro del fichero `settings/prod.py`

```
ALLOWED_HOSTS = [  
    "127.0.0.1",  
    "localhost",  
    "my-icard-saas-env.eba-kvwxg4uf.us-west-2.elasticbeanstalk.com",  
]
```

Ilustración 43-3: Fichero django.config dentro de “.ebextensions”

Realizado por: Cuenca A.; Llinin F., 2022.

c) AWS RDS PostgreSQL

Se debe acceder a la consola del entorno donde está ejecutándose la aplicación e ir a la configuración de Base de datos (Database). Para crear una nueva base de datos se necesitan los siguientes datos:

- **Motor de base de datos (Engine):** postgres
- **Versión de base de datos (Engine version):** 12.9
- **Clase de instancia (Instance class):** La recomendación de AWS es usar db.t2.micro, sin embargo, para el caso de “My iCard” por la cantidad de paquetes de terceros se necesita usar una instancia db.t2.medium.
- **Almacenamiento (Storage):** 5GB
- **Nombre de usuario (Username):** Nombre de usuario para acceder a la base de datos.
- **Contraseña (Password):** Contraseña para acceder a la base de datos.

Para empezar, se debe crear un nuevo directorio raíz con el nombre “.ebextensions”. Dentro de este, se debe definir lo descrito en la **Ilustración 44-3**.

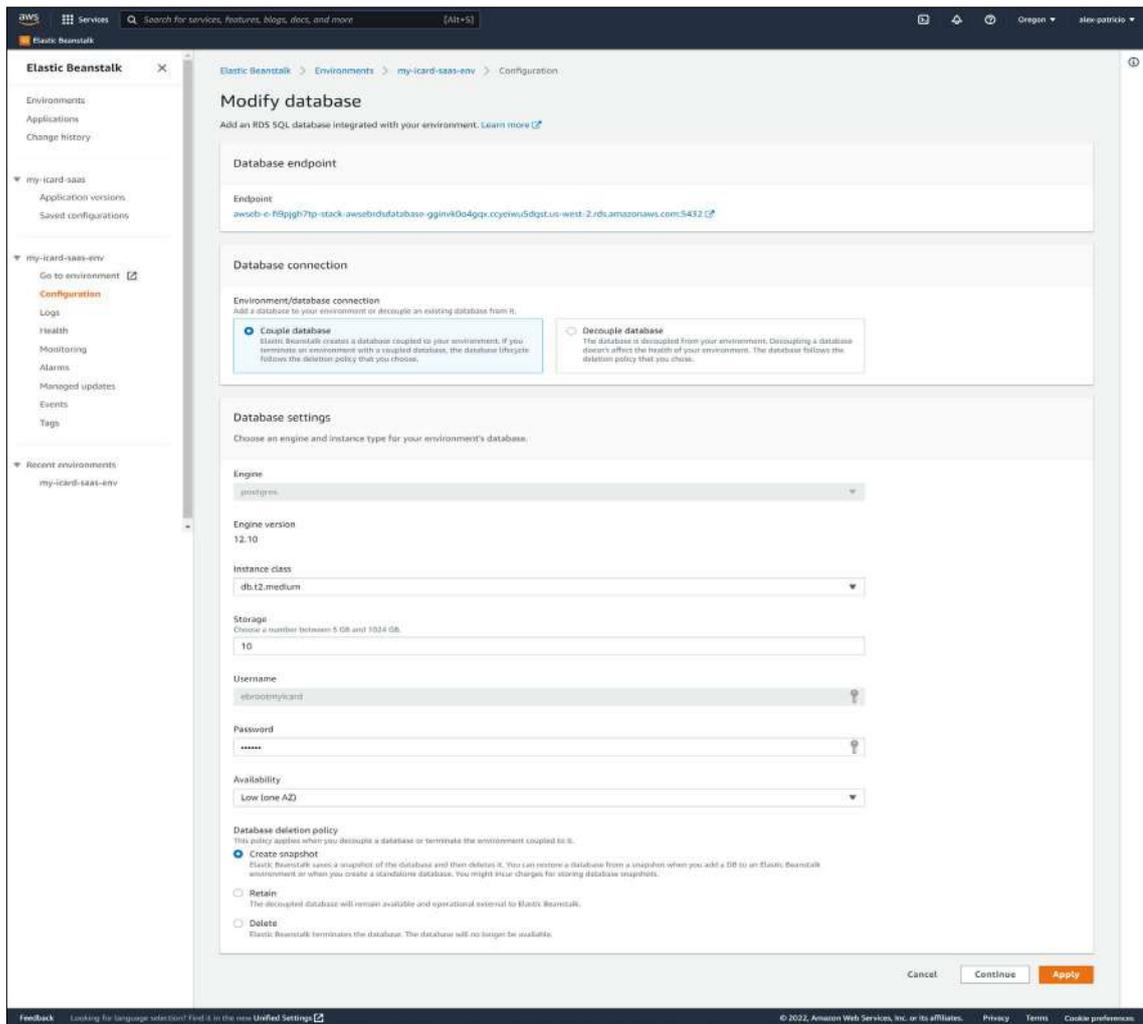


Ilustración 44-3: Configuración de base de datos en AWS RDS

Realizado por: Cuenca A.; Llinin F., 2022.

Para que la aplicación pueda reconocer la nueva base de datos creada, se debe configurar la nueva base de datos como lo indica la **Ilustración 45-3**.

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql_psycopg2",
        "NAME": env("RDS_DB_NAME"),
        "USER": env("RDS_USERNAME"),
        "PASSWORD": env("RDS_PASSWORD"),
        "HOST": env("RDS_HOSTNAME"),
        "PORT": env("RDS_PORT"),
    }
}
```

Ilustración 45-3: Fichero django.config dentro de “.ebextensions”

Realizado por: Cuenca A.; Llinin F., 2022.

d) HTTPS con AWS Certificate Manager

Para continuar con esta sección se debe contar con un dominio previamente registrado. Para este caso, se ha escogido usar AWS Route 53 y adquirir el dominio “my-icard-saas.link”.

i) Certificado SSL/TLS

Para esto se debe abrir la consola de AWS Certificate Manager. Se debe solicitar un certificado. Para este caso se debe solicitar un certificado público. A continuación, se debe ingresar el dominio y escoger el método de validación DNS. El proceso de validación del certificado empieza una vez se haya comprobado de que el dominio es de su propiedad. Para ello se debe preparar el nombre y valor del CNAME del dominio. En caso de comprobar la propiedad el estado cambia de “Pendiente de validación” (Pending validation) a “Emitido” (Issued).

ii) Balanceador de carga para redirigir HTTPS

Para configurar un nuevo balanceador de carga se debe crear un nuevo fichero “.securelistener-clb.config” dentro de “.ebextensions”.

```
option_settings:
  aws:elb:listener:443:
    SSLCertificateId: arn:aws:acm:us-west-2: ...
    ListenerProtocol: HTTPS
    InstancePort: 80
```

Ilustración 46-3: Fichero “.securelistener-clb.config” dentro de “.ebextensions”

Realizado por: Cuenca A.; Llinin F., 2022.

iii) Host permitidos

Ahora se debe volver a configura-r el dominio configurado en esta sección en ALLOWED_HOSTS:

```
ALLOWED_HOSTS = [
    ...
    "my-icard.netlify.app",
    ...
]
```

Ilustración 47-3: Configuración ALLOWED_HOSTS dentro de settings/prod.py

Realizado por: Cuenca A.; Llinin F., 2022.

Finalmente, se debe redirigir todo el tráfico de HTTP a HTTPS. Para esto, se debe configurar Apache como host proxy. Para ello, se debe añadir un nuevo fichero dentro de settings/prod.py con el contenido de la Ilustración 48-3.

```
SECURE_PROXY_SSL_HEADER = ("HTTP_X_FORWARDED_PROTO", "https")
SESSION_COOKIE_SECURE = True
CSRF_COOKIE_SECURE = True
```

Ilustración 48-3: Configuración SSL dentro de settings/prod.py

Realizado por: Cuenca A.; Llinin F., 2022.

Ahora, en el navegador se puede ir a la dirección https://. Con la configuración anterior, al ir a la dirección http://, automáticamente debe acceder a un dominio seguro. Un ejemplo correcto de configuración se muestra en la **Ilustración 49-3**.

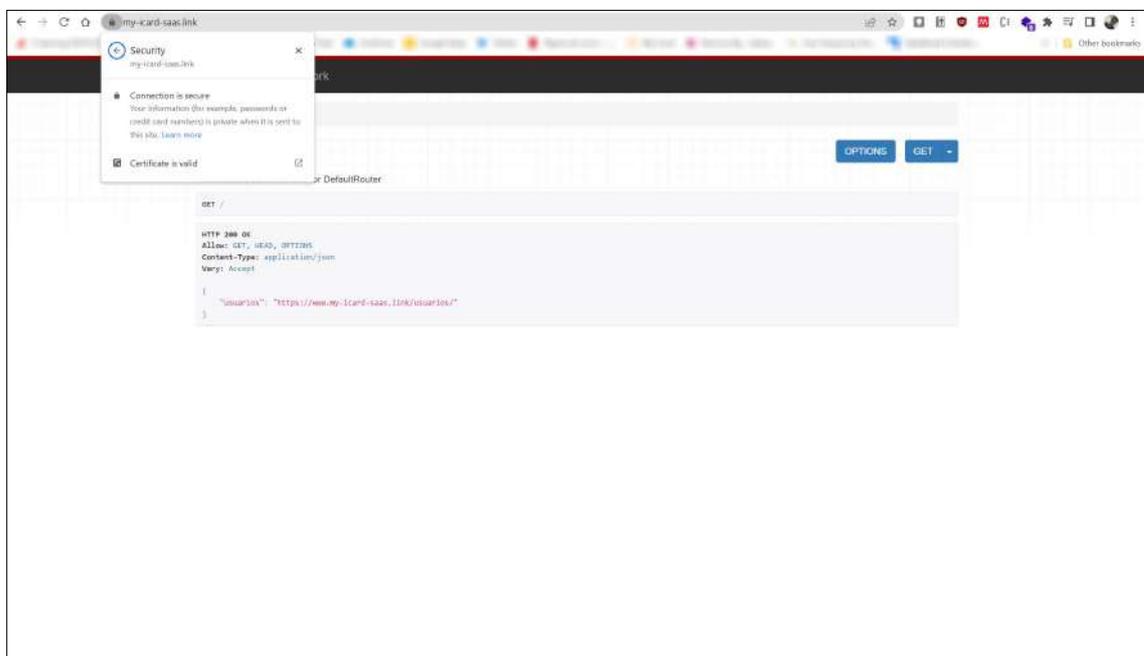


Ilustración 49-3: Configuración de base de datos en AWS RDS

Realizado por: Cuenca A.; Llinin F., 2022.

3.4. Metodología para la determinación de la eficiencia de desempeño

En esta sección se da a conocer los indicadores para medir la eficiencia de desempeño de “My iCard”, mediante la aplicación de fichas permite valorar cada subcaracterística de la eficiencia como se muestra en las **Tabla 13-3**, **Tabla 14-3** y **Tabla 15-3**.

Tabla 12-3: Indicadores para medir la eficiencia

Variable	Indicador	Descripción
Eficiencia de desempeño	Comportamiento temporal	Es la habilidad del software para ser modificado de manera efectiva de acuerdo con las necesidades del usuario.
	Utilización de recursos	

Realizado por: Cuenca A.; Llinin F., 2022.

Tabla 13-3: Ficha para el comportamiento en el tiempo

Característica	Eficiencia de desempeño
Subcaracterísticas	Comportamiento en el tiempo
Métrica	Tiempo de respuesta
Propósito	Conocer el tiempo de respuesta en que genera cada pedido.
Medición, fórmula	T = Tiempo en minutos (calculado o simulado) T= (Tiempo en obtener el resultado) - (Tiempo en introducir el comando)
Interpretación	Cuanto menor sea el tiempo mejor ($0 < T$)
Tiempo de escala	Segundos
Tipo de medida	T = Tiempo
Fuente de medición	Sistema My iCard
ISO/IEC 12207 SLCP	Validación
Audiencia	Desarrolladores, clientes

Realizado por: Cuenca A.; Llinin F., 2022.

Tabla 14-3: Ficha para la utilización de recursos

Característica	Eficiencia de desempeño
Subcaracterísticas	Utilización de recursos
Métrica	Uso de memoria RAM
Propósito	Conocer cuánto de memoria RAM utiliza cada tarea
Medición, fórmula	X=tamaño en MB
Interpretación	El menor tamaño es mejor
Tiempo de escala	Ratio
Fuente de medición	Tamaño estimado de la funcionalidad
ISO/IEC 12207 SLCP	Verificación y Validación
Audiencia	Desarrolladores.

Realizado por: Cuenca A.; Llinin F., 2022.

Tabla 15-3: Ficha para la utilización de recursos

Característica	Eficiencia de desempeño
Subcaracterísticas	Utilización de recursos
Métrica	Uso de procesador
Propósito	Conocer cuánto de procesador utiliza cada tarea
Medición, fórmula	X= % (Porcentaje)
Interpretación	El menor tamaño es mejor
Tiempo de escala	Ratio
Fuente de medición	Porcentaje estimado de la funcionalidad
ISO/IEC 12207 SLCP	Verificación y Validación
Audiencia	Desarrolladores

Realizado por: Cuenca A.; Llinin F., 2022.

3.4.1. Población y muestra

3.4.1.1. Población y muestra de la eficiencia de desempeño

Con el objetivo de medir la eficiencia de desempeño del sistema My iCard se consideró toda la población de todos los procesos manuales que realiza el administrador para la toma de pedidos, los mismos que fueron considerados como muestra, los cuales se detallan a continuación:

- Agregar una mesa
- Realizar un pedido
- Supervisar estados del pedido
- Generar la cuenta de un pedido
- Reporte historial de pagos

Considerando que son varios procesos y la población de cada uno de estos tiende al infinito, se procede a la realización de una técnica de muestreo estratificado con población infinita, como consecuencia se divide la población en subgrupos.

Para encontrar el tamaño de la muestra se calcula mediante la fórmula de muestra con población infinita usando un índice de confianza de 95% ($z=1.96$), la probabilidad de éxito y de fracaso al 50% ($p = 0.5$; $q= 0.5$) y un error del 5% ($e= 0.05$).

$$n = \frac{Z^2 * p * q}{e^2}$$

$$n = \frac{(1.96)^2 * 0.5 * 0.5}{(0.05)^2}$$

$$n = 384.16$$

$$n \approx 384$$

El tamaño de la muestra de cada proceso se lo halla mediante la aplicación de un muestreo estratificado como se observa en la **Tabla 16-3**. El campo de frecuencia hace referencia al número aproximado de veces que se realiza el proceso en un día.

Tabla 16-3: Muestreo estratificado para las operaciones

Funcionalidad	Frecuencia	Porcentaje	Muestra
Agregar una mesa	1	8%	31
Realizar un pedido	40	28%	108
Supervisar estados del pedido	40	28%	108
Generar la cuenta de un pedido	40	28%	108
Reporte historial de pagos diarios	1	8%	31
Total	122	100%	384

Realizado por: Cuenca A.; Llinin F., 2022.

3.4.2. Planteamiento de la hipótesis

El objetivo de la hipótesis planteada en el Trabajo de Integración Curricular es evaluar la eficiencia en el desempeño del sistema My iCard para los procesos mencionados en la población de estudio, por lo que se calcula el tiempo promedio de los procesos. Dado esto se proponen las siguientes hipótesis:

H0: El tiempo de respuesta del sistema es mayor o igual al tiempo promedio del proceso manual del negocio.

H1: El tiempo de respuesta del sistema es menor al tiempo promedio del proceso manual del negocio.

CAPITULO IV

4. RESULTADOS

En presente capítulo tiene como propósito realizar un análisis, evaluar y describir los resultados obtenidos para la eficiencia en el desempeño del sistema “My iCard”.

4.1. Eficiencia en el desempeño

Para la medición de la eficiencia de desempeño se evaluó el comportamiento temporal y la utilización de recursos.

4.1.1. *Comportamiento temporal*

La métrica usada para realizar esta medición fue el tiempo de respuesta.

4.1.1.1. *Tiempo de respuesta*

Para la medición de esta métrica se consideró la variable X como el tiempo de respuesta de una funcionalidad del sistema empujando la técnica de observación mediante el uso de fichas y un cronómetro. Por consecuencia se obtendrá una comparativa de tiempo del proceso manual y el proceso automatizado para verificar si existe una disminución de tiempo en la atención del cliente.

Para el presente estudio se ha considerado los procesos que el nutricionista lleva a cabo manualmente como se observa en la **Tabla 1-4**, donde se encuentra descrito el tiempo que conlleva cada una de las funcionalidades.

Tabla 1-4: Tiempo manual del proceso de pedidos

Funcionalidad	Tiempo en minutos (min)
Agregar una mesa	5 min
Realizar un pedido	15 min

Supervisar estados del pedido	4 min
Generar la cuenta de un pedido	6 min
Reporte historial de pagos diarios	12 min
Total	42 min

Realizado por: Cuenca A.; Llinin F., 2022.

Se evidencia que el tiempo empleado por los trabajadores de la pizzería en cada proceso no es muy notable pero la actividad en la que radica mayor tiempo en la de realizar un pedido, debido a que por las normas de bioseguridad este proceso es más complejo que antes.

Tiempo de respuesta con el sistema web My iCard

Mediante el uso del sistema se pretende reducir los tiempos empleados en el proceso de pedidos de la pizzería. Como se puede observar en la **Tabla 2-4**, se detalla los diferentes tiempos en cada una de las funcionalidades del sistema.

Tabla 2-4: Tiempo de respuesta con My iCard

Funcionalidad	Tiempo en segundos (s)	Tiempo en minutos (min)
Agregar una mesa	37,6 s	0,63 min
Realizar un pedido	153,8 s	2,56 min
Supervisar estados del pedido	36,8 s	0,61 min
Generar la cuenta de un pedido	54,5 s	0,91 min
Reporte historial de pagos diarios	127,7 s	2,13 min
Total	410,4 s	6,84 min

Realizado por: Cuenca A.; Llinin F., 2022.

De acuerdo con los datos obtenidos al realizar un análisis del tiempo de respuesta del sistema My iCard ocupa alrededor de 6.84 minutos la realización de todo el proceso de pedidos del cliente.

4.1.1.2. Comparación de los tiempos del proceso.

Con el propósito de presentar los resultados de las actividades realizadas en la pizzería, tanto en el proceso manual como en el proceso automatizado, se presenta una comparativa como se puede observar en la **Tabla 3-4**, en donde ambas medidas se encuentran en minutos.

Tabla 3-4: Comparativa de resultados entre el proceso manual y automatizado

Funcionalidad	Proceso manual en minutos (min)	Proceso automatizado en minutos (min)
Agregar una mesa	5 min	0,63 min
Realizar un pedido	15 min	2,56 min
Supervisar estados del pedido	4 min	0,61 min
Generar la cuenta de un pedido	8 min	0,91 min
Reporte historial de pagos diarios	12 min	2,13 min
Total	44 min	6,84 min

Realizado por: Cuenca A.; Llinin F., 2022.

En base a los resultados obtenidos en la Tabla 3-4, el proceso manual emplea un total de 44 minutos, mientras que, en el proceso automatizado se utiliza un total de 6.84 minutos, evidenciando la disminución de 37.16 minutos. En la **Ilustración 1-4** se puede observar los tiempos de cada una de las funcionalidades donde la disminución de tiempo por cada proceso es notoria.

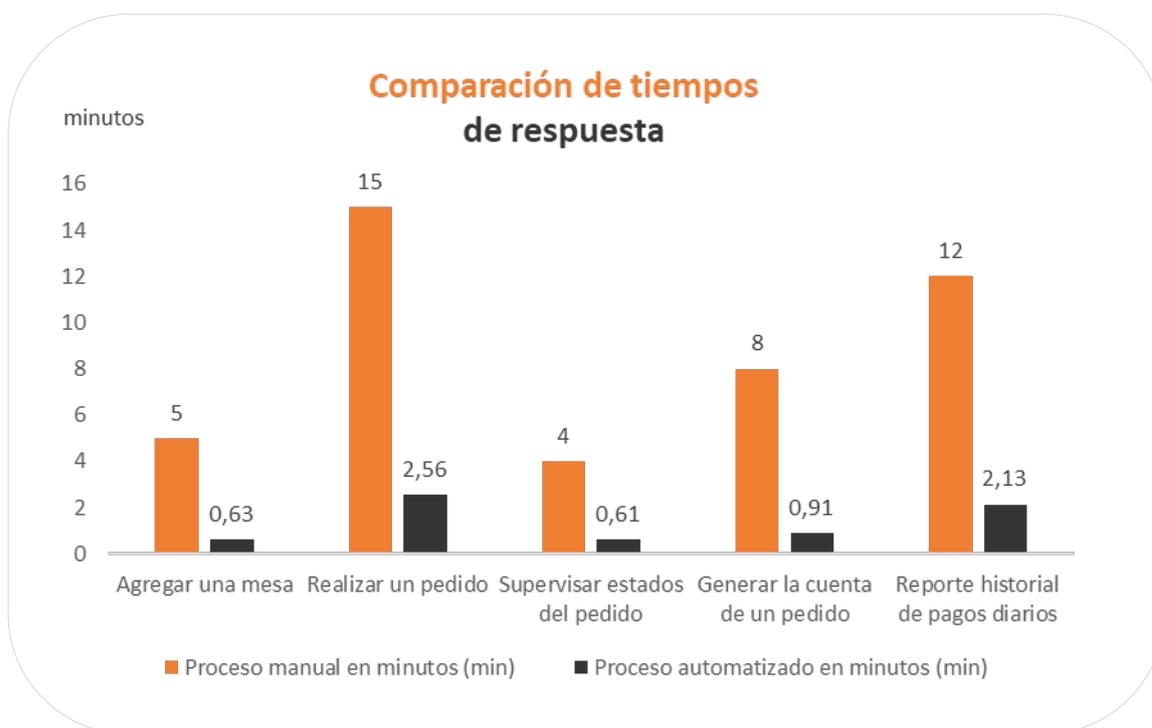


Ilustración 1-4: Comparativa de tiempos por funcionalidades

Realizado por: Cuenca A.; Llinin F., 2022.

Una vez observado las diferencias de los tiempos, en la **Ilustración 2-4** se observa el porcentaje de los tiempos obtenidos, tomando en cuenta que se brinda la valoración de 100% al proceso

manual y de esta forma se obtiene el porcentaje del tiempo del proceso automatizado dando un valor de 15,5 %. De acuerdo con los resultados obtenidos, con el uso de My iCard el proceso automatizado se reduce en 84.45 % con respecto al tiempo con el proceso manual que efectuaba la pizzería.

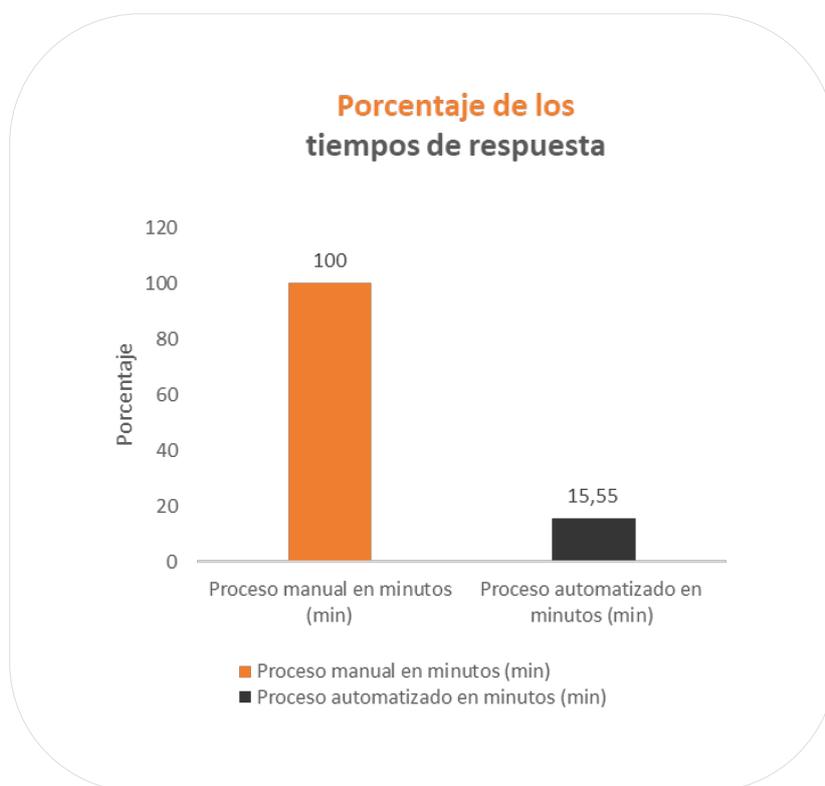


Ilustración 2-4: Porcentaje de los tiempos de respuesta

Realizado por: Cuenca A.; Llinin F., 2022

Acorde con nuestras hipótesis planteadas, con los datos obtenidos como se observa en la **Tabla 4-4**, se realizó el uso del estadístico t de Student para muestras relacionadas donde se calculó la diferencia en los tiempos.

Tabla 4-4: Cálculo de la diferencia entre los tiempos

Nro.	Proceso Manual	Proceso Automatizado	d
1	5 min	0,63 min	4,4 min
2	15 min	2,56 min	12,4 min
3	4 min	0,61 min	3,4 min
4	8 min	0,91 min	7,1 min
5	12 min	2,13 min	9,9 min

Realizado por: Cuenca A.; Llinin F., 2022.

Mediante la aplicación de la fórmula $S_d = \sqrt{\frac{(d_i - \bar{d})^2}{n-1}}$ el resultado de la desviación estándar fue de $S_d = 3,77009$.

Con este valor calcularemos el estadístico t mediante el empleo de la fórmula $t = \frac{\bar{d}}{s_d/\sqrt{n}}$ siendo \bar{d} la media de las diferencias y n el número de los datos en este caso 5, en donde se obtuvo el valor $t = 4,40796719$. Para el análisis de este estadístico se tomó en cuenta los grados de libertad que tienen valor 4 y el nivel de significancia establecido al 5%. Estos datos ayudaran a calcular el valor crítico mediante la fórmula $t_{(1-\alpha),(n-1)}$ en donde se obtiene un valor de 2.131847.

Como se observa en la **Ilustración 3-4**, después de realizar la prueba t de Student con muestras relacionadas al tener el valor del estadístico $t=4.4079$ y el valor crítico de 2.1318, se rechaza la hipótesis nula y se acepta la hipótesis alterna, teniendo como resultado que el tiempo de respuesta del sistema es menor al tiempo promedio del proceso manual del negocio.



Ilustración 3-4: Prueba t de Student

Realizado por: Cuenca A.; Llinin F., 2022.

4.1.2. Utilización de recursos

Para realizar la evaluación de esta subcaracterística se consideró las métricas de uso de memoria RAM y uso de procesador, mismos que se evaluaron en cada uno de los procesos. Se ha

considerado que mientras las funcionalidades ocupen menos recursos será muy bueno, las unidades consideradas son megabytes (MB) para el uso de memoria RAM y el porcentaje (%) con respecto al uso del procesador.

4.1.2.1. Uso de memoria RAM

Para la evaluación de esta métrica se consideró la variable $X = \text{MB}$ que es la medida que se obtiene de una determinada funcionalidad. Se ha dividido la medición en dos grupos; el primero por parte del servidor en AWS y el segundo en el equipo del cliente. Además, como se observa en la **Tabla 5-4** hace referencia a un indicador de evaluación del uso de la memoria RAM utilizado en el trabajo de (Gómez, Arcos-Medina y Pástor 2020).

Tabla 5-4: Indicadores de evaluación uso de memoria RAM

Calificación %	Intervalo de RAM	Valor cualitativo
100%	[0-150]MB	Excelente
90%	[151-250]MB	Muy Bueno
75%	[251-350]MB	Bueno
50%	[351-450]MB	Aceptable
20%	[451-550]MB	Regular
0%	[551-∞]MB	Malo

Fuente: (Gómez, Arcos-Medina y Pástor 2020)

Realizado por: Cuenca A.; Llinin F., 2022.

4.1.2.2. Uso del procesador (CPU)

Para la evaluación de esta métrica se consideró la variable $X = \text{%(Porcentaje)}$ que es la medida para realizar una funcionalidad. Se ha dividido la medición en dos grupos; el primero por parte del servidor en AWS y el segundo en el equipo del cliente. Además, como se observa en la **Tabla 6-4** hace referencia a un indicador de evaluación del uso de procesador utilizado en el trabajo de (Gómez, Arcos-Medina y Pástor 2020).

Tabla 6-4: Indicadores de evaluación uso de CPU

Calificación %	Porcentaje	Valor cualitativo
100%	[0-0,5]	Excelente
90%	[0,6-1,5]	Muy Bueno
75%	[1,6-2,5]	Bueno
50%	[2,6-3,5]	Aceptable
20%	[3,6-4,5]	Regular

0%	[4,6-∞]	Malo
----	---------	------

Fuente: (Gómez, Arcos-Medina y Pástor 2020)

Realizado por: Cuenca A.; Llinin F., 2022.

Mediciones realizadas del uso de RAM y procesador

Debido a que en este proyecto se ha propuesto una solución cloud del tipo SaaS, la primera toma de métricas se la realiza desde el servidor en el que la aplicación desplegada se encuentra corriendo dentro de Amazon Web Services. Una vez se han configurado los entornos de despliegue y la consola de administración de Amazon Cloud Watch, se han podido determinar las medidas que se presentan a continuación.

a) Agregar una mesa

Peticiones HTTP de código 2XX



Ilustración 4-4: Peticiones HTTP 2XX cuando se agrega una mesa

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.



Ilustración 5-4: Utilización de memoria RAM y CPU cuando se agrega una mesa.

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

El promedio de memoria RAM utilizada para agregar una mesa es de: 326 MB. Por otro lado, el porcentaje de CPU utilizado es de: 0.83%

b) Realizar un pedido

Peticiones HTTP de código 2XX



Ilustración 6-4: Peticiones HTTP 2XX cuando se realiza un pedido.

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.



Ilustración 7-4: Utilización de memoria RAM y CPU cuando se realiza un pedido.

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

El promedio de memoria RAM utilizada es: 340 MB. Por otro lado, el porcentaje de CPU utilizado es de: 0.82%

c) Supervisar estados del pedido

Para la toma de medida de este procedimiento se la ha realizado desde las perspectivas del lado del administrador (tablero de mandos) y desde la aplicación del cliente.

i) Desde el tablero de mandos

Peticiones HTTP de código 2XX



Ilustración 8-4: Peticiónes HTTP 2XX cuando se supervisa estados del pedido

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.



Ilustración 9-4: Utilización de memoria RAM y CPU cuando se supervisa estados del pedido desde el tablero de mandos.

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

ii) Desde la aplicación del cliente

Peticiónes HTTP de código 2XX



Ilustración 10-4: Peticiónes HTTP 2XX cuando se supervisa estados del pedido desde la aplicación del cliente.

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

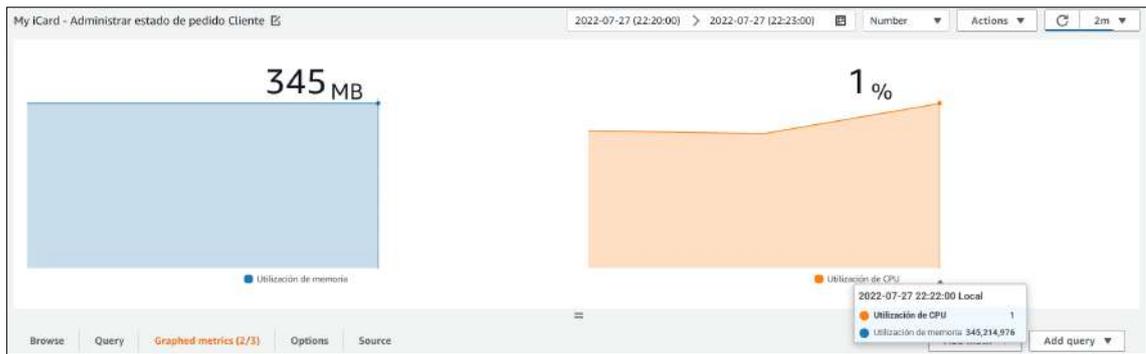


Ilustración 11-4: Utilización de memoria RAM y CPU cuando se supervisa estados del pedido desde la aplicación del cliente.

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

Entre ambos enfoques tenemos que el promedio de memoria RAM utilizada es: 345 MB. Por otro lado, el porcentaje de CPU utilizado es: 0.91%

d) Generar la cuenta de un pedido

Para la toma de medida de este procedimiento se la ha realizado desde las perspectivas del lado del administrador (tablero de mandos) y desde la aplicación del cliente.

- i) Desde el tablero de mandos

Peticiones HTTP de código 2XX



Ilustración 12-4: Peticiones HTTP 2XX cuando se genera la cuenta de un pedido

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.



Ilustración 13-4: Utilización de memoria RAM y CPU cuando se genera la cuenta de un pedido

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

i) Desde la aplicación del cliente

Peticiones HTTP de código 2XX



Ilustración 14-4: Peticiones HTTP 2XX cuando se genera la cuenta de un pedido

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.



Ilustración 15-4: Utilización de memoria RAM y CPU cuando se genera la cuenta de un pedido

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

Entre ambos enfoques tenemos que el promedio de memoria RAM utilizada es: 344,5 MB. Por otro lado, el porcentaje de CPU utilizado es: 0.83%

e) **Reporte historial de pagos diarios**

Peticiones HTTP de código 2XX

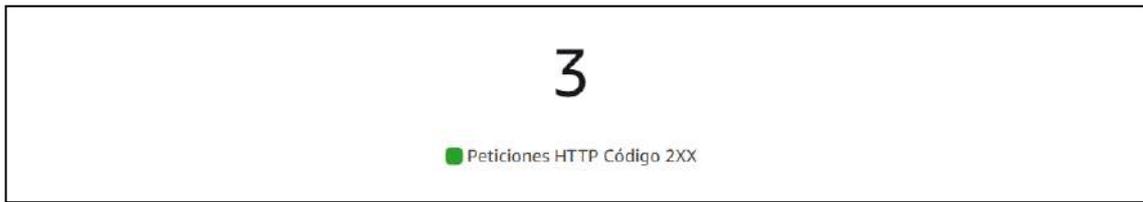


Ilustración 16-4: Peticiones HTTP 2XX cuando se extrae el reporte historial de pagos diarios

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

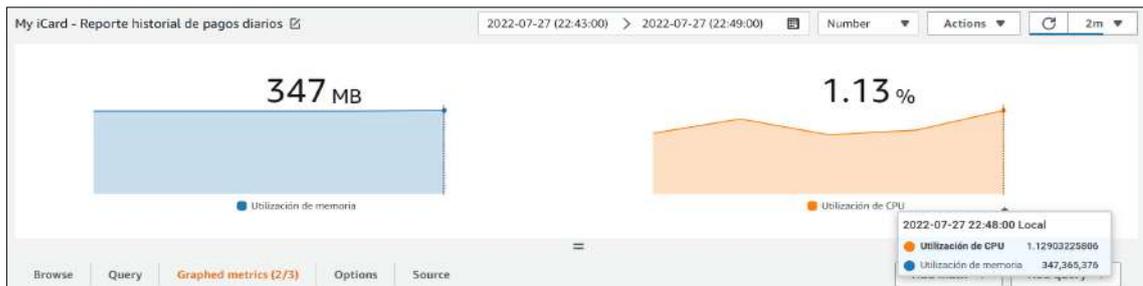


Ilustración 17-4: Utilización de memoria RAM y CPU cuando se extrae el reporte historial de pagos diarios.

Fuente: (Amazon CloudWatch, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

El promedio de memoria RAM utilizada es: 347 MB. Por otro lado, el porcentaje de CPU utilizado es de: 1.13%

Como se observa en la **Tabla 7-4**, se tiene los resultados consolidados tanto en la utilización de RAM como en la utilización del procesador CPU de parte de nuestro servidor en AWS

Tabla 7-4: Resultados de las mediciones en el servidor

Funcionalidad	Uso del procesador (%)	Uso de la RAM (MB)
Agregar una mesa	0,83 %	326 MB
Realizar un pedido	0,82 %	340 MB
Supervisar estados del pedido	0,91 %	345 MB
Generar la cuenta de un pedido	0,83 %	344,5 MB
Reporte historial de pagos diarios	1,13 %	347 MB
Promedio	0,904 %	340,5 MB

Realizado por: Cuenca A.; Llinin F., 2022

En la **Ilustración 18-4** se puede observar el consumo de RAM por parte del servidor en cada una de las funcionalidades mientras que en la **Ilustración 19-4** se observa el consumo del

procesador en las funcionalidades mediadas, que indica que la funcionalidad de reporte del historial de pagos diario consume mayor RAM y procesador .



Ilustración 18-4: Utilización de RAM en el servidor de AWS por cada funcionalidad

Realizado por: Cuenca A.; Llinin F., 2022



Ilustración 19-4: Utilización del CPU en el servidor de AWS por cada funcionalidad

A nivel de servidor tras el análisis de la métrica uso del procesador, se obtuvo un promedio de 0,904% de las tareas efectuadas, de acuerdo con la **Tabla 6-4**, el valor se ubica en la escala de [0,6-1,5] alcanzando una valoración de Muy Bueno con un porcentaje de 90%. En el análisis de la métrica de uso de memoria RAM se obtuvo un promedio de 340,5 MB, acorde con la **Tabla 5-4**, el valor se ubica en la escala de [251-350]MB alcanzando una calificación de Bueno con un porcentaje de 75%.

Complementando la primera toma de mediciones, la segunda se la realiza desde el computador donde se encuentre ejecutando la aplicación. Para esto se han determinado 3 navegadores que se encuentren ejecutando únicamente la aplicación web My iCard. Mediante el uso del administrador de tareas de Windows se obtiene el PID de la tarea donde se está ejecutando la ventana. Con el PID del paso anterior se procede a buscarlo en el software de aplicación para medición de procesos Process Explorer desde el cual se puede obtener una vista detallada del proceso.

Obtención de PID de la ventana donde se ejecuta My iCard

Navegador Microsoft Edge

El indicador 1 corresponde a la ventana donde se ejecuta My iCard Administradores y el indicador 2 corresponde a la ventana donde se ejecuta My iCard Clientes.

Name	Status	PID	35% CPU	79% Memory
Microsoft Edge (26)			0.4%	580.9 MB
Browser		45844	0.1%	53.9 MB
Console Window Host		59300	0%	0.6 MB
Console Window Host		57872	0%	0.6 MB
Console Window Host		30756	0%	0.6 MB
Crashpad		38436	0%	1.0 MB
Extension: IDM Integration Module		51392	0%	19.2 MB
Extension: Kaspersky Password Manager		21968	0%	14.4 MB
Extension: Kaspersky Protection		42384	0%	17.2 MB
Extension: Microsoft Power Automate		5880	0%	13.9 MB
GPU Process		43448	0.1%	105.6 MB
Kaspersky Native Messaging Server (32 bit)		54216	0%	3.6 MB
Native Messaging Server for Kaspersky Password ...		15820	0%	7.3 MB
Prerender: New tab (1)		60724	0%	59.7 MB
Spare Renderer		59288	0%	7.7 MB
Subframe: https://stripe.com/ (1)		20028	0.1%	60.0 MB
Subframe: https://stripe.network/ (1)		14552	0%	63.2 MB
Tab: My iCard		50200	0%	55.2 MB
Tab: My iCard		3344	0%	48.2 MB
Utility: Network Service		7212	0%	10.8 MB
Utility: Search Indexer Service		33748	0%	1.1 MB
Utility: Semantic Encoder Service		16212	0%	24.7 MB
Utility: Storage Service		21604	0%	1.9 MB
Web Automation Native Message Host		47288	0.1%	9.7 MB
Windows Command Processor		36524	0%	0.2 MB
Windows Command Processor		64440	0%	0.3 MB

Ilustración 20-4: PID de la ventana del navegador Microsoft Edge donde se ejecuta el sistema administrador

Fuente: (Administrador de tareas de Windows 11, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

Navegador Opera

El indicador 1 corresponde a la ventana donde se ejecuta My iCard Administradores y el indicador 2 corresponde a la ventana donde se ejecuta My iCard Clientes.

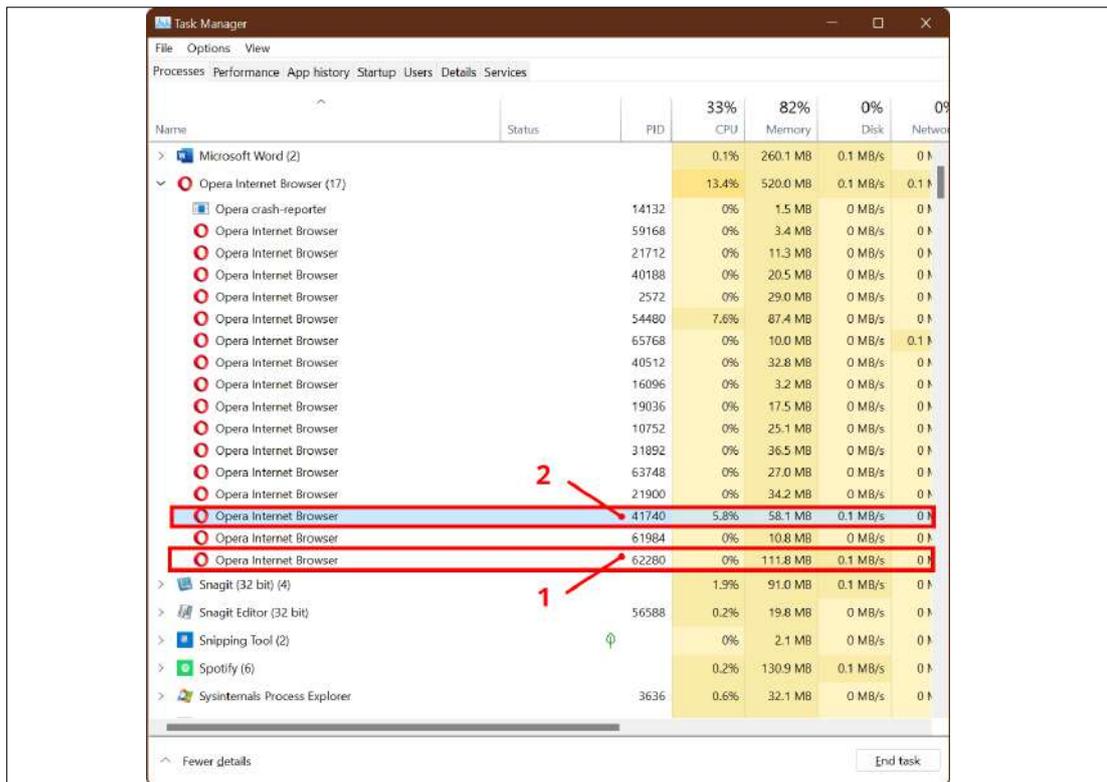


Ilustración 21-4: PID de la ventana del navegador Opera Internet donde se ejecuta el sistema administrador

Fuente: (Administrador de tareas de Windows 11, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

Navegador Opera

El indicador 1 corresponde a la ventana donde se ejecuta My iCard Administradores y el indicador 2 corresponde a la ventana donde se ejecuta My iCard Clientes.

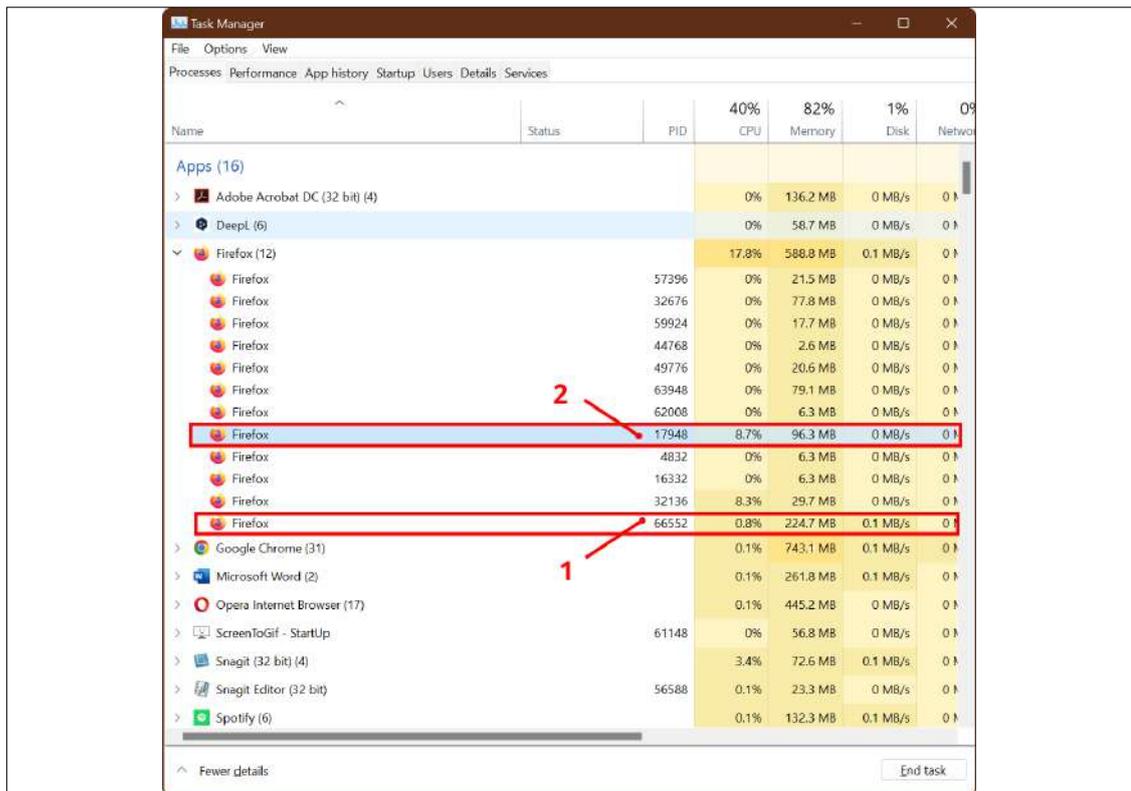


Ilustración 22-4: PID de la ventana del navegador Mozilla Firefox donde se ejecuta el sistema administrador

Fuente: (Administrador de tareas de Windows 11, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

a) Agregar una mesa

Los valores obtenidos tras la medición de esta funcionalidad en los navegadores propuestos como se observa en la **Ilustración 23-4** se han consolidado en la **Tabla 8-4** para una mejor comprensión.

Tabla 8-4: Utilización de memoria y CPU cuando se agrega una mesa

Navegador	Métrica	
	Utilización de procesador (%)	Utilización de memoria (MB)
Microsoft Edge	2.35 %	117.4 MB
Opera Internet	2.86 %	88.2 MB
Mozilla Firefox	2.04 %	213.6 MB

Realizado por: Cuenca A.; Llinin F., 2022.

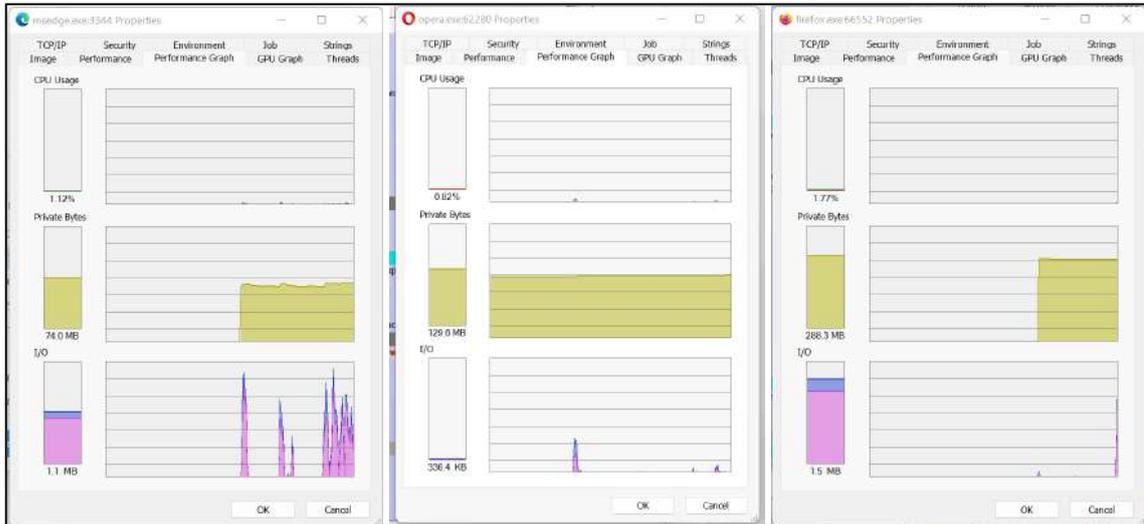


Ilustración 23-4: Utilización de memoria y CPU cuando se agrega una mesa

Fuente: (Process Explorer, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

b) Realizar un pedido

Los valores obtenidos tras la medición de esta funcionalidad en los navegadores propuestos como se observa en la **Ilustración 24-4** se han consolidado en la **Tabla 9-4** para una mejor comprensión.

Tabla 9-4: Utilización de memoria y CPU cuando se realiza un pedido

Navegador	Métrica	
	Utilización de procesador (%)	Utilización de memoria (MB)
Microsoft Edge	2.35 %	117.4 MB
Opera Internet	2.86 %	88.2 MB
Mozilla Firefox	2.04 %	213.6 MB

Realizado por: Cuenca A.; Llinin F., 2022.

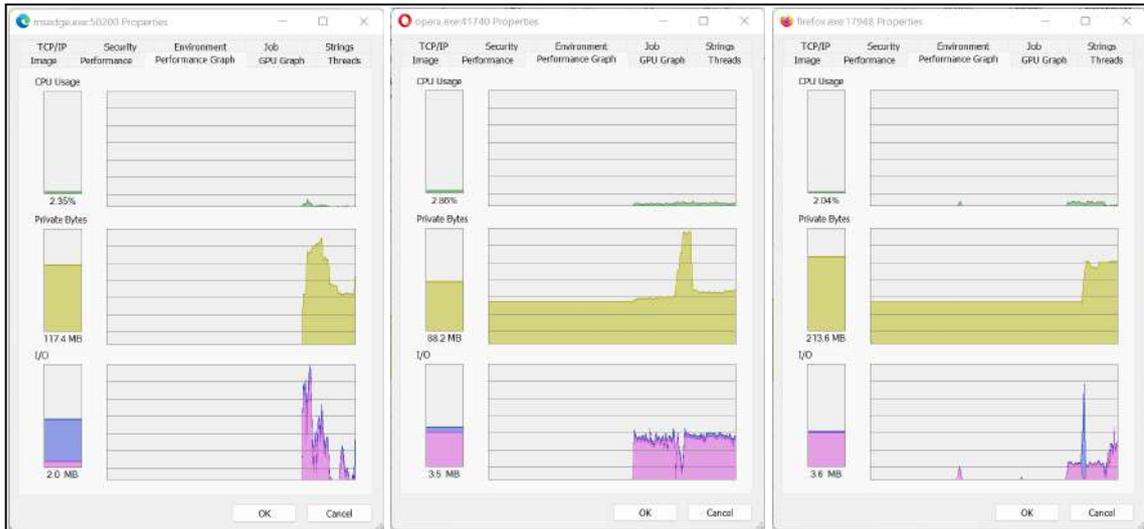


Ilustración 24-4: Utilización de memoria y CPU cuando se realiza un pedido

Fuente: (Process Explorer, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

c) Supervisar estados del pedido

Para la toma de medida de este procedimiento se la ha realizado desde las perspectivas del lado del administrador (tablero de mandos) y desde la aplicación del cliente.

iii) Desde el tablero de mandos

Los valores obtenidos tras la medición de esta funcionalidad en los navegadores propuestos como se observa en la **Ilustración 25-4** se han consolidado en la **Tabla 10-4** para una mejor comprensión.

Tabla 10-4: Utilización de memoria y CPU cuando se supervisa estados del pedido desde el tablero de mandos

Navegador	Métrica	
	Utilización de procesador (%)	Utilización de memoria (MB)
Microsoft Edge	1.74 %	82.6 %
Opera Internet	1.82 %	128.0 %
Mozilla Firefox	1.69 %	310.8 %

Realizado por: Cuenca A.; Llinin F., 2022.

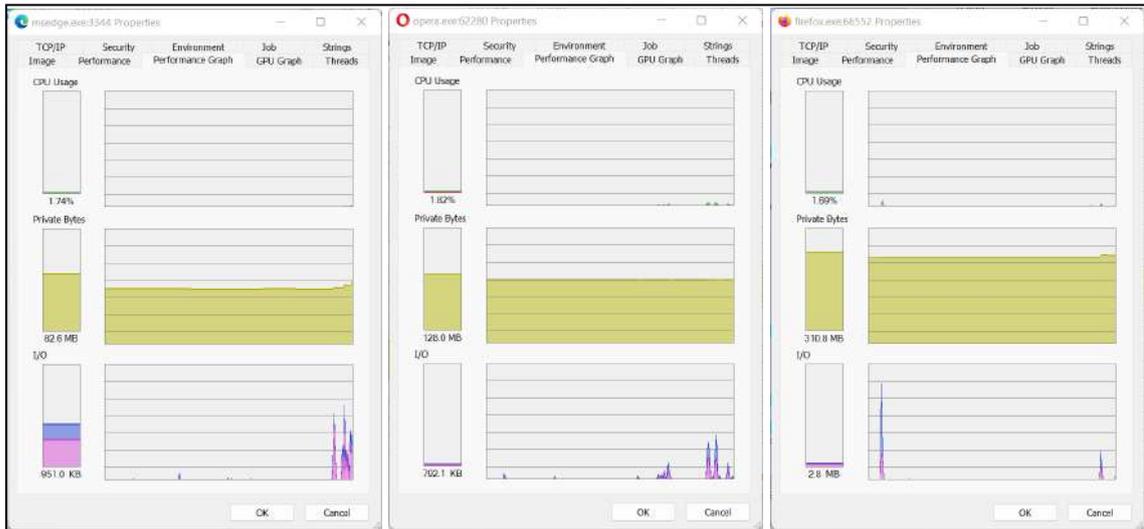


Ilustración 25-4: Utilización de memoria y CPU cuando se supervisa estados del pedido desde el tablero de mandos

Fuente: (Process Explorer, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

iv) Desde la aplicación del cliente

Los valores obtenidos tras la medición de esta funcionalidad en los navegadores propuestos como se observa en la **Ilustración 26-4** se han consolidado en la **Tabla 11-4** para una mejor comprensión.

Tabla 11-4: Utilización de memoria y CPU al supervisar estados del pedido desde la aplicación del cliente

Navegador	Métrica	
	Utilización de procesador (%)	Utilización de memoria (MB)
Microsoft Edge	2.65 %	97.7 MB
Opera Internet	1.76 %	81.4 MB
Mozilla Firefox	2.25 %	130.7 MB

Realizado por: Cuenca A.; Llinin F., 2022.

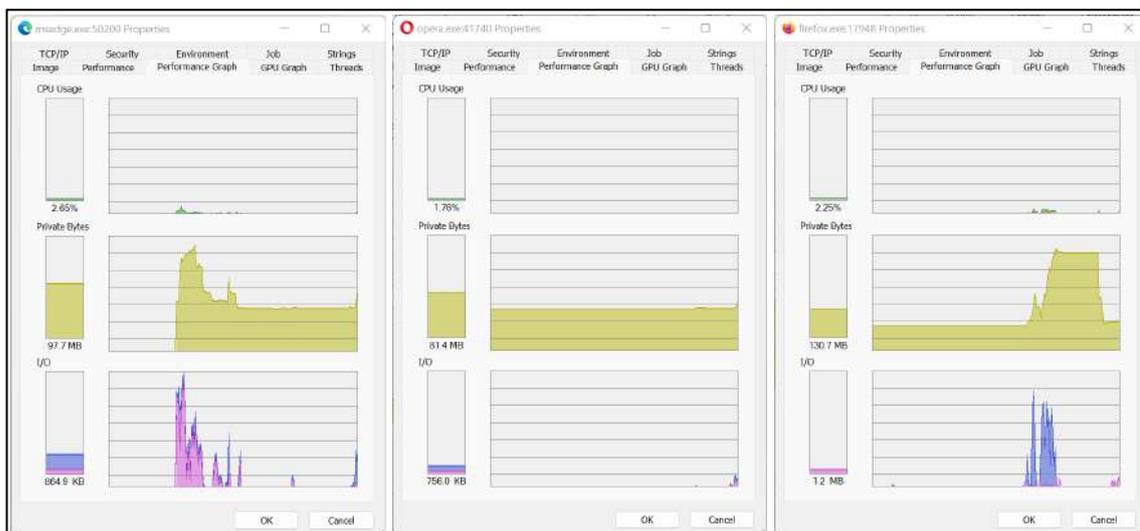


Ilustración 26-4: Utilización de memoria y CPU al supervisar estados del pedido desde la aplicación del cliente

Fuente: (Process Explorer, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

d) Generar la cuenta de un pedido

Para la toma de medida de este procedimiento se la ha realizado desde las perspectivas del lado del administrador (tablero de mandos) y desde la aplicación del cliente.

- i) Desde el tablero de mandos

Los valores obtenidos tras la medición de esta funcionalidad en los navegadores propuestos como se observa en la **Ilustración 27-4** se han consolidado en la **Tabla 12-4** para una mejor comprensión.

Tabla 12-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido

Navegador	Métrica	
	Utilización de procesador (%)	Utilización de memoria (MB)
Microsoft Edge	0.26 %	102.2 MB
Opera Internet	0.27 %	128.7 MB
Mozilla Firefox	0.16 %	161.6 MB

Realizado por: Cuenca A.; Llinin F., 2022.

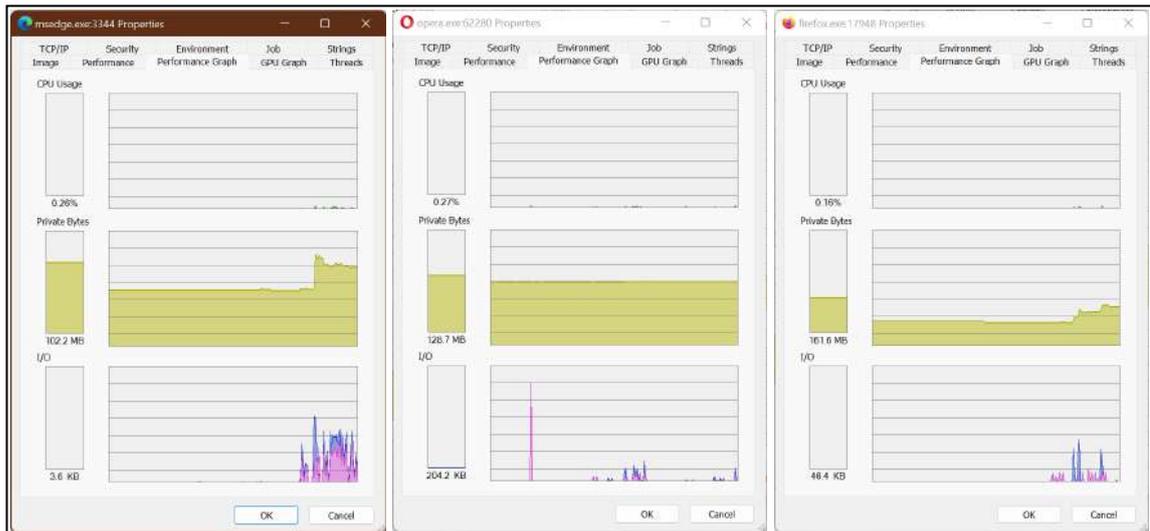


Ilustración 27-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido

Fuente: (Process Explorer, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

ii) Desde la aplicación del cliente

Los valores obtenidos tras la medición de esta funcionalidad en los navegadores propuestos como se observa en la **Ilustración 28-4** se han consolidado en la **Tabla 13-4** para una mejor comprensión.

Tabla 13-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido

Navegador	Métrica	
	Utilización de procesador (%)	Utilización de memoria (MB)
Microsoft Edge	1.97 %	116.8 MB
Opera Internet	1.50 %	128.9 MB
Mozilla Firefox	1.27 %	229.7 MB

Realizado por: Cuenca A.; Llinin F., 2022.

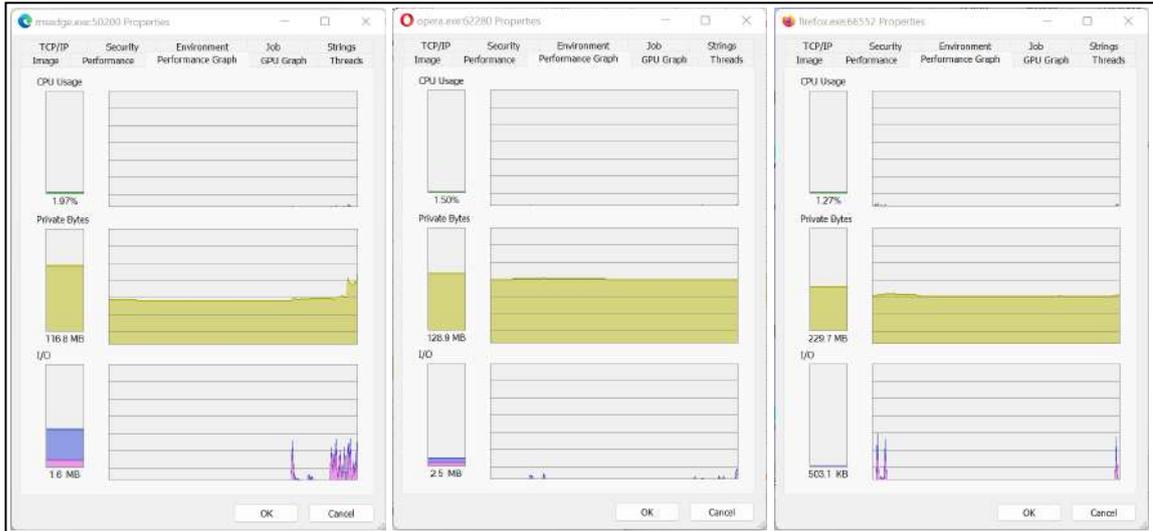


Ilustración 28-4: Utilización de memoria y CPU cuando se genera la cuenta de un pedido

Fuente: (Process Explorer, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

e) Reporte historial de pagos diarios

Los valores obtenidos tras la medición de esta funcionalidad en los navegadores propuestos como se observa en la **Ilustración 29-4** se han consolidado en la **Tabla 14-4** para una mejor comprensión.

Tabla 14-4: Utilización de memoria y CPU al extraer el reporte historial de pagos diarios

Navegador	Métrica	
	Utilización de procesador (%)	Utilización de memoria (MB)
Microsoft Edge	2.12 %	122.3 MB
Opera Internet	2.14 %	129.4 MB
Mozilla Firefox	2.49 %	238.3 MB

Realizado por: Cuenca A.; Llinin F., 2022.

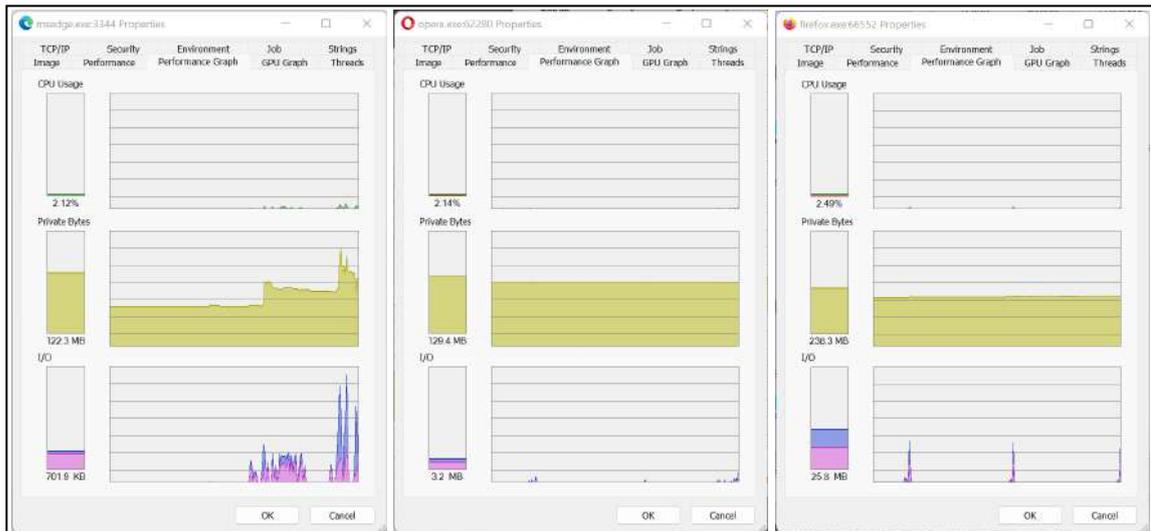


Ilustración 29-4: Utilización de memoria y CPU al extraer el reporte historial de pagos diarios

Fuente: (Process Explorer, 2022)

Realizado por: Cuenca A.; Llinin F., 2022.

Como se observa en la **Tabla 15-4** se realiza una tabla comparativa de los navegadores en la utilización de RAM con los resultados de las mediciones anteriores, muestra un promedio de las funcionalidades para cada navegador. En la **Ilustración 30-4** existen variaciones en cada una de las tareas acorde con los navegadores usados, pero para tener una idea más clara la **Ilustración 31-4** el navegador Opera Internet consume menos RAM con un promedio de 107.86 MB tras la realización de las funciones medidas. Se toma el dato del navegador Opera para el análisis de la métrica uso de memoria RAM, se obtuvo un promedio de 107,86 MB, acorde con la **Tabla 5-4** del indicador, el valor se ubica en la escala de [0-150]MB alcanzando una calificación de Excelente con un porcentaje de 100%.

Tabla 15-4: Resultados de las mediciones de RAM en distintos navegadores

Funcionalidad	Microsoft Edge	Opera Internet	Mozilla Firefox
Agregar una mesa	117,4 MB	88,2 MB	213,6 MB
Realizar un pedido	117,4 MB	88,2 MB	213,6 MB
Supervisar estados del pedido	90,15 MB	104,7 MB	220,75 MB
Generar la cuenta de un pedido	109,5 MB	128,8 MB	195,65 MB
Reporte historial de pagos diarios	122,3 MB	129,4 MB	238,3 MB
Promedio	111,35 MB	107,86 MB	216,38 MB

Realizado por: Cuenca A.; Llinin F., 2022.

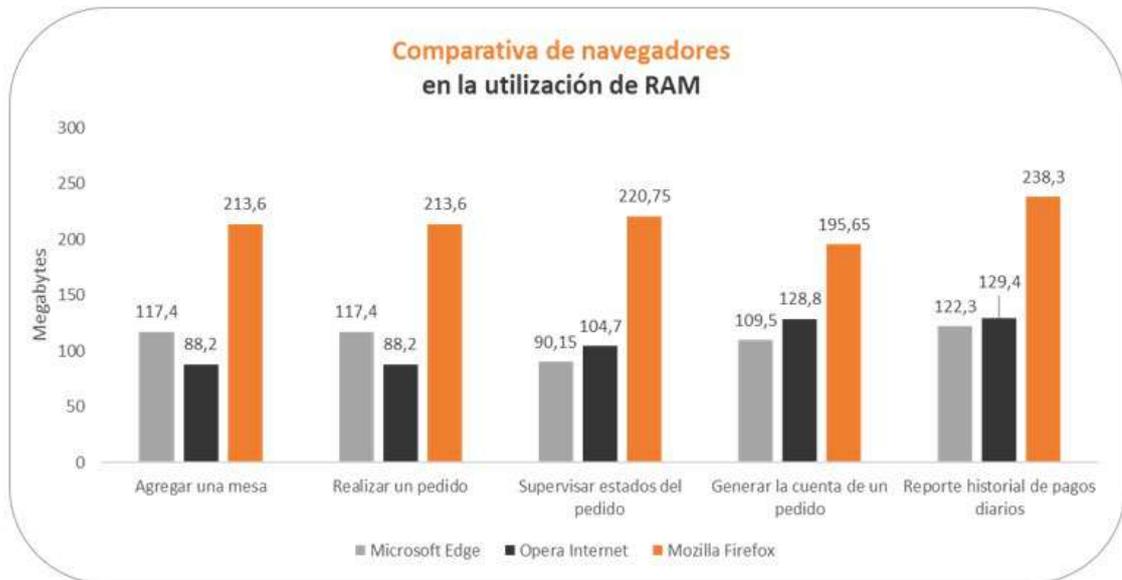


Ilustración 30-4: Comparativa del uso de RAM en diferentes navegadores

Realizado por: Cuenca A.; Llinin F., 2022

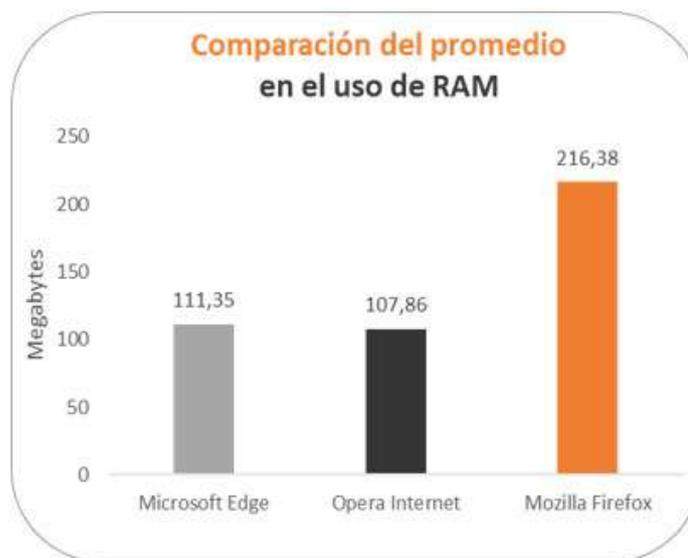


Ilustración 31-4: Comparativa del promedio de uso de RAM

Realizado por: Cuenca A.; Llinin F., 2022

La **Tabla 16-4** corresponde a una comparativa de los navegadores en la utilización del procesador con los resultados de las mediciones anteriores, muestra un promedio de las funcionalidades para cada navegador. La **Ilustración 32-4** muestra la existencia de variaciones en cada una de las tareas acorde con los navegadores usados, pero para tener una idea más clara la **Ilustración 33-4** el navegador Opera Internet utiliza en menor porcentaje el procesador con un promedio de 1.95 % tras la realización de las funciones medidas. Se toma el dato del navegador Opera para el análisis de la métrica uso de uso del procesador, se obtuvo un

promedio de 1,95 %, acorde con la **Tabla 6-4** del indicador, el valor se ubica en la escala de [1,6-2,5] alcanzando una calificación de Bueno con un porcentaje de 75%.

Tabla 16-4: Resultados de las mediciones de RAM en distintos navegadores

Funcionalidad	Microsoft Edge	Opera Internet	Mozilla Firefox
Agregar una mesa	2,35 %	2,06 %	2,84 %
Realizar un pedido	2,35 %	2,86 %	2,04 %
Supervisar estados del pedido	2,20 %	1,79 %	1,97 %
Generar la cuenta de un pedido	1,12 %	0,89 %	0,72 %
Reporte historial de pagos diarios	2,12 %	2,14 %	2,49 %
Promedio	2,03 %	1,95 %	2,01 %

Realizado por: Cuenca A.; Llinin F., 2022.

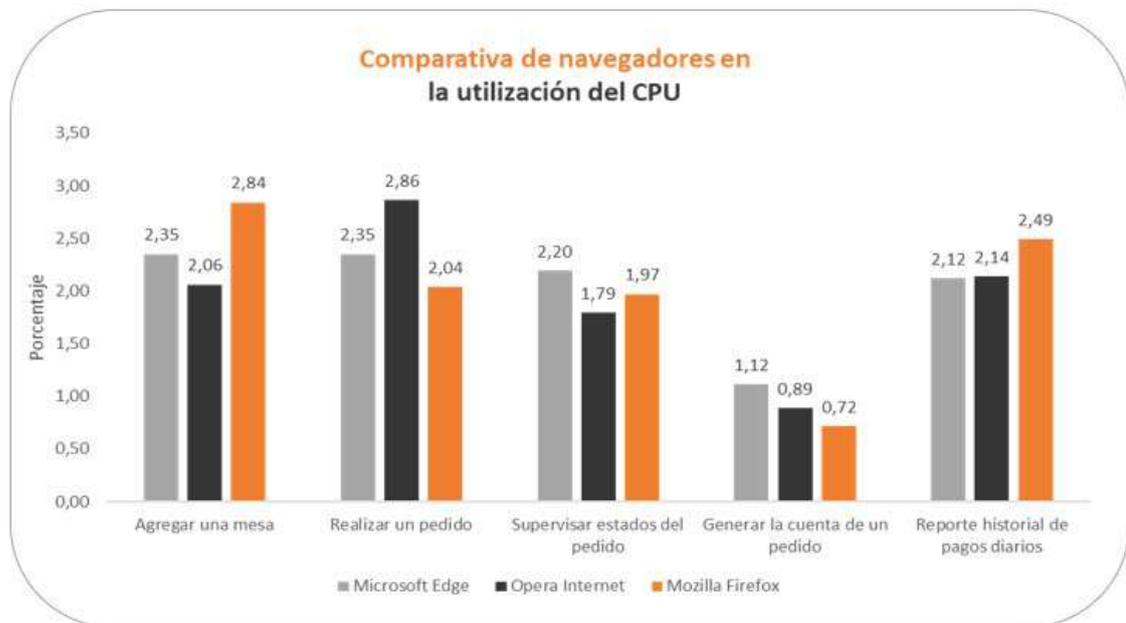


Ilustración 32-4: Comparativa del uso del procesador en diferentes navegadores

Realizado por: Cuenca A.; Llinin F., 2022.

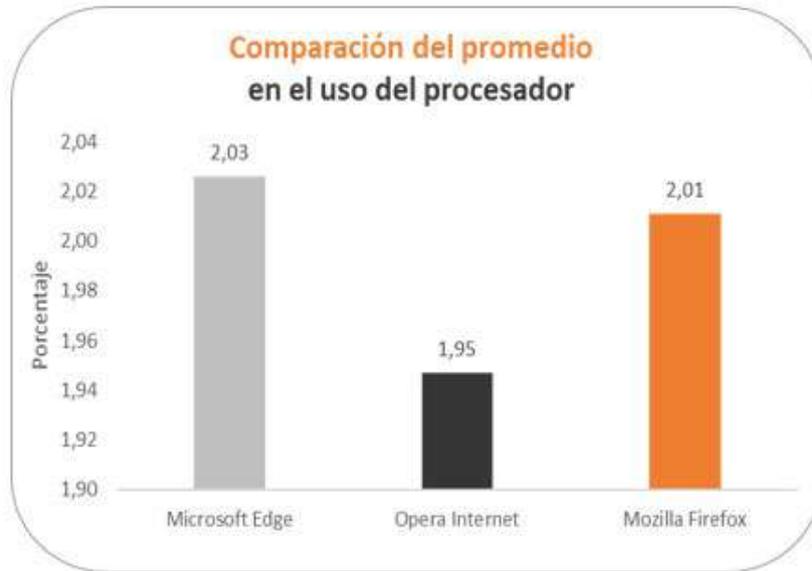


Ilustración 33-4: Comparativa del promedio de uso del procesador

Realizado por: Cuenca A.; Llinin F., 2022.

Resultados de la eficiencia de desempeño

Finalmente, de los resultados obtenidos de la característica de la eficiencia de desempeño se ha considerado una ponderación en cada una de las subcaracterística como el comportamiento temporal y la utilización de recursos, adicionalmente en la utilización de recursos se realiza el análisis tanto a nivel de servidor como a nivel del cliente por lo que en la **Tabla 17-4** vemos estos valores de manera individual mientras que en la **Tabla 18-4** se observa el consolidado global y el detalle de las respectivas asignaciones de las Subcaracterísticas

Tabla 17-4: Ponderaciones de las subcaracterísticas a nivel de servidor y cliente

Comportamiento temporal 50%	Utilización de recursos 50 %			
	Utilización de memoria - server	Utilización de procesador - cliente	Utilización de memoria - cliente	Utilización de procesador - cliente
Tiempo de respuesta				
47,2	9,4	11,3	12,5	9,4

Realizado por: Cuenca A.; Llinin F., 2022.

Tabla 18-4: Resultados de la eficiencia de desempeño

Comportamiento temporal 50%	Utilización de recursos 50 %		Eficiencia de desempeño 100%
Tiempo de respuesta	Utilización de memoria	Utilización de procesador	Total
47,2	21,9	20,6	89,7

Realizado por: Cuenca A.; Llinin F., 2022.

En la **Ilustración 32-4** se detalla la eficiencia de desempeño total del sistema, una vez medidas las dos subcaracterísticas de la ISO/IEC 25010:2011, en efecto se muestra una eficiencia del 90% del sistema, esto se debe a que no se obtuvo un buen puntaje en el uso de RAM por parte del servidor y uso del procesador por parte del cliente, por esta razón no se alcanzó el porcentaje total sin embargo nos encontramos con un resultado Muy Bueno.



Ilustración 34-4: Resultado de la eficiencia de desempeño

Realizado por: Cuenca A.; Llinin F., 2022.

CONCLUSIONES

- El gerente actualmente maneja un proceso tradicional al momento de realizar pedidos en su pizzería, donde además se ve involucrado más personal como meseros quienes receptan los pedidos y cajera quien saca el costo de un pedido de forma manual, estos procesos fueron representados en un diagrama basado en el estándar BPMN 2.0
- El modelo SaaS permite alojar software directamente en la nube, sin que se necesite de un ordenador físico que lo mantenga, brindando mayor conectividad desde cualquier lugar que cuente con conexión a internet, sin embargo, solo el proveedor del servicio puede añadir funcionalidades y modificar los códigos fuente. AWS es la mayor plataforma a nivel mundial que brinda servicios para este tipo de modelo.
- My iCard se desarrolló mediante la utilización de la metodología Kanban para la gestión y el ciclo de vida de software, en un total de 5 meses contando con 28 requisitos, permitiendo dar cumplimiento a los 7 módulos que requiere el gerente.
- Se determinó que el nivel de eficiencia de desempeño del sistema My iCard es del 90%, en base a la medición del comportamiento temporal, utilización de memoria RAM y uso del procesador tanto de cliente como del servidor. Mediante la automatización del sistema se logró reducir 37.16 minutos en cuanto a la atención del cliente y se determinó que el navegador Opera Internet usa los recursos de manera más eficiente.

RECOMENDACIONES

- Los usuarios del sistema My iCard pueden utilizar el navegador Opera Internet para tener un mejor rendimiento de la aplicación debido a que este ha tenido el menor porcentaje de uso de recursos como RAM y procesador.
- Para un mejor manejo de My iCard los usuarios deben visualizar el manual de usuario, el mismo que presenta información necesaria para interactuar con el sistema fácilmente.
- La reutilización de código es una clara ventaja que se tiene en la programación, para futuros trabajos se recomienda el uso de la API desarrollada en el presente estudio que permite la creación de un front end que consume los servicios RESTful desarrollados.
- Los servicios ofrecidos por AWS (Amazon Web Services) para implementar software tipo SaaS, brindan gran variedad de paquetes de herramientas y una excelente relación costo-beneficio a favor de los desarrolladores.
- Agregar el certificado SSL (Secure Sockets Layer) al dominio para poder consumir de manera efectiva los datos alojados en el Servidor de AWS, así también como para aumentar la protección frente ataques, asegurar la privacidad de los datos y proporcionar la confianza y credibilidad del sistema para los usuarios.

BIBLIOGRAFÍA

AGGARWAL, Sanchit. *Modern Web-Development using ReactJS*. 2018. vol. 5, no. 1, pp. 5.

AWS. *AWS Elastic Beanstalk - Guía para desarrolladores*. 2019. pp. 1165.

BANCO BILBAO VIZCAYA ARGENTARIA, S.A. . *Qué es el método Kanban y por qué funciona en la programación de software*. *BBVA API_Market* [en línea]. 2019. [Consulta: 14 enero 2022]. Disponible en: <https://www.bbvaapimarket.com/es/mundo-api/que-es-el-metodo-kanban-y-por-que-funciona-en-la-programacion-de-software/>.

BERMEJO, Marcos, FLORIT, Marc. y SEDÓ, Ramón. *Producción multimedia*. [en línea]. 2011. [Consulta: 15 enero 2022]. Disponible en: http://cv.uoc.edu/annotation/ebc1adfc61836d7205ad7dde343367b5/603266/PID_00236250/PID_00236250.html.

BRAVO, Ángel. *El SaaS y el Cloud-Computing: una opción innovadora para tiempos de crisis*. 2009. no. 1, pp. 5.

CHÁVEZ, Susana., MARTÍN, Adriana, RODRÍGUEZ, Nelson, MURAZZO, María. y VALENZUELA, Adriana. *Metodología AGIL para el desarrollo SaaS. XIV Workshop de Investigadores en Ciencias de la Computación* [en línea]. 2012. S.l.: s.n., [Consulta: 14 enero 2022]. ISBN 978-950-766-082-5. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/18977>.

CHICAIZA, Johanna. *Desarrollo de un sistema vía Web para automatizar los pedidos y reservaciones de comidas en la Parrilla “Don Carlos” del cantón Milagro* [en línea]. 2020. S.l.: Universidad Estatal de Milagro. [Consulta: 14 enero 2022]. Disponible en: <http://repositorio.unemi.edu.ec/handle/123456789/5150>.

DE BONIS, German. *¿Qué es una «comanda» de restaurante? Guía Completa*. [en línea]. 2019. [Consulta: 15 enero 2022]. Disponible en: <https://germandebonis.com/la-comanda-de-restaurante-normas-procedimientos-y-ordenes/>.

DIGITÉ. *¿Qué Es Kanban? Una Descripción General Del Método Kanban*. [en línea]. 2021 [Consulta: 30 mayo 2022]. Disponible en: <https://www.digite.com/es/kanban/que-es-kanban/>.

DIGNAN, Larry. *Top cloud providers: AWS, Microsoft Azure, and Google Cloud, hybrid, SaaS players*. *ZDNet* [en línea]. 2021. [Consulta: 2 mayo 2022]. Disponible en: <https://www.zdnet.com/article/the-top-cloud-providers-of-2021-aws-microsoft-azure-google-cloud-hybrid-saas/>.

EGUILUZ, Javier. *Introducción a JavaScript*. [en línea]. 2009. [Consulta: 15 enero 2022]. Disponible en: <https://uniwebsidad.com/libros/javascript?from=librosweb>.

GALLEGO, Jesús. *Manual práctico de restaurante*. 2002. S.l.: Editorial Paraninfo. ISBN 978-84-283-2512-7.

GARCÍA. *Pedidos y comandas: organización del servicio entre comedor y cocina*. *EN LA COCINA Magazine* [en línea]. 2018. [Consulta: 15 enero 2022]. Disponible en: <https://enlacocina.telemesa.es/gestion-administracion-restaurantes/pedidos-y-comandas-organizacion-del-servicio-entre-comedor-y-cocina/>.

GOMEZ, Ingrith. *¿Qué es SaaS? Ventajas y desventajas.* <https://www.crehana.com> [en línea]. 2022. [Consulta: 2 mayo 2022]. Disponible en: <https://www.crehana.com/ec/blog/desarrollo-web/que-es-saas/>.

GÓMEZ, Omar, ARCOS-MEDINA, Gloria y PÁSTOR, Danilo. *Application of Genetic Algorithms Technique in the Generation of Academic Schedules.* 2020. *KnE Engineering*, pp. 150-165. ISSN 2518-6841. DOI 10.18502/keg.v5i1.5927.

GURENDO, Dmitry. *Kanban Model and Process for Software Development: Principles, Steps, Methodology.* *XB Software* [en línea]. 2015. [Consulta: 30 mayo 2022]. Disponible en: <https://xbsoftware.com/blog/software-development-life-cycle-sdlc-all-about-kanban/>.

IBM. *¿Qué es la gestión de pedidos?* [en línea]. 2021. [Consulta: 15 enero 2022]. Disponible en: <https://www.ibm.com/co-es/topics/order-management>.

IONOS. *SaaS: ventajas del software a demanda.* *IONOS Digitalguide* [en línea]. 2019. [Consulta: 15 enero 2022]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-saas/>.

LENDÍNEZ, Laura. *Kanban. Metodología para aumentar la eficiencia de los procesos.* 2019. *3C Tecnología. Glosas de innovación aplicadas a la pyme*, pp. 30-41. ISSN 2254-4143.

LIAWATIMENA, Suryadiputra. *Django Web Framework Software Metrics Measurement Using Radon and Pylint.* 2018. *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*. S.l.: s.n., pp. 218-222. DOI 10.1109/INAPR.2018.8627009.

LLERENA, Erika y GUILLEN, Kleber. *APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE CONTROL ACADÉMICO UNACH.* 2020. Riobamba: Universidad Nacional de Chimborazo.

MACÍAS, Ana. *Conoce las ventajas y desventajas de software como servicio (SaaS) para clínicas | Cirrus Blog.* [en línea]. 2018. [Consulta: 18 abril 2022]. Disponible en: <https://www.getcirrus.com/blog/conoce-las-ventajas-y-desventajas-de-software-como-servicio-saas-para-clinicas>.

MALDONADO, Fernando. *SaaS: Un mercado en plena expansión.* 2009. *IDC*, pp. 11.

MENDOZA, *DESARROLLO DE UN SISTEMA WEB, QUE PERMITA ADMINISTRAR PEDIDOS, APLICADA A LA DISTRIBUIDORA REPUESTOS CÓRDOVA.* S.l.: Universidad de Guayaquil.

MICROSOFT Visual Studio: IDE y Editor de código para desarrolladores de software y Teams. [en línea]. [Consulta: 15 enero 2022]. Disponible en: <https://visualstudio.microsoft.com/es/>.

MIRALLES, *Cloud computing y protección de datos.* pp. 11.

NÚÑEZ ESPINOZA, *Implementación de un prototipo de Software como Servicio (SaaS) para pequeñas y medianas empresas* [en línea]. Quito: Escuela Politécnica Nacional. [Consulta: 15 enero 2022]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/7108>.

ORBIT LOGISTICS. *Automatización de pedidos: optimice la logística de su empresa.* *Portal VMI* [en línea]. 2020. [Consulta: 15 enero 2022]. Disponible en:

<https://blog.portalvmi.com.br/es/automatizacion-de-pedidos-optimice-la-logistica-de-su-empresa-2/>.

PALOS-SANCHEZ, ARENAS, y MARIANO, The adoption of cloud computing technology (SaaS): Effects of technological complexity vs training and support. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, vol. 2017, pp. 89-105. DOI 10.17013/risti.22.89-105.

PARAMARINI, Analysis and Design of User Interface and User Experience (UI / UX) E-Commerce Website PT Pentasada Andalan Kelola Using Task System Centered Design (TCSD) Method. *2019 Fourth International Conference on Informatics and Computing (ICIC)*. S.l.: s.n., pp. 1-8. DOI 10.1109/ICIC47613.2019.8985854.

PÉREZ, Guía comparativa de metodologías ágiles. En: Accepted: 2012-09-18T08:56:03Z [en línea [Consulta: 15 enero 2022]]. Disponible en: <https://uvadoc.uva.es/handle/10324/1495>.

POSTGRESQL PostgreSQL: About. [en línea]. [Consulta: 2 mayo 2022]. Disponible en: <https://www.postgresql.org/about/>.

RAWAT, Prateek y MAHAJAN, Archana. ReactJS: A Modern Web Development Framework. 2020. vol. 5, no. 11, pp. 5.

REACT. Introduction - Semantic UI React. [en línea]. 2022. [Consulta: 2 mayo 2022]. Disponible en: <https://react.semantic-ui.com/>.

RESCEANU, Ionut, REȘCEANU, Cristina. y SIMIONESCU, Sabin. SaaS solutions for small-medium businesses: Developer's perspective on creating new SaaS products. *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*. 2014. S.l.: s.n., pp. 140-144. DOI 10.1109/ICSTCC.2014.6982405.

ROCHE ¿Qué es la metodología Kanban? | Deloitte España. *Deloitte Spain* [en línea]. [Consulta: 15 enero 2022]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-metodologia-kanban.html>.

ROMERO, AGNETTI y MEDRANO, RETOS EN LA CADENA DE SUMINISTRO DE ALIMENTOS ASOCIADOS A LA PANDEMIA DE COVID-19. pp. 9.

SARASWAT, y TRIPATHI, R. Cloud Computing: Comparison and Analysis of Cloud Service Providers-AWs, Microsoft and Google. *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*. S.l.: s.n., pp. 281-285. DOI 10.1109/SMART50582.2020.9337100.

SEMANTIC UI. [en línea]. [Consulta: 2 mayo 2022]. Disponible en: <https://semantic-ui.com/>.

TAPIA, ÁLVAREZ ZURITA, y ANDRADE, Desarrollo organizacional como estrategia de modernización de la calidad del servicio de alimentos y bebidas. *Visionario Digital*, vol. 3, no. 2.2, pp. 103-119. ISSN 2602-8506. DOI 10.33262/visionariodigital.v3i2.2.618.

ZAMORA, El desarrollo organizacional como generador de un cambio significativo en las organizaciones de la sociedad civil. *Almenara: revista extremeña de ciencias sociales*, no. 10, pp. 87-109. ISSN 1889-6286, 2172-7538.

ANEXOS

ANEXO A: Documento validación de entrega del Manual Técnico y de Usuario



PIZZERIA D' BAGGIO

Riobamba, 2022-09-05

Srs.

Alex Cuenca, Francisco Llinin

ESTUDIANTES DE LA ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

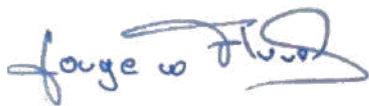
Presentes.

De mi consideración:

Reciban un cordial saludo, el motivo de la presente es para validar la entrega del Manual Técnico y Manual de Usuario, con motivo de la realización de su Trabajo de Integración Curricular en mi establecimiento, indicando que la documentación fue entregada de manera física y electrónica.

Particular que comunico para los fines pertinentes.

Atentamente;



Jorge Washington Flores Córdor

GERENTE DE PIZZERIA D' BAGGIO



ANEXO B: Evolución mensual del Tablero Kanban

Para el management del proyecto se utilizó la metodología Kanban por lo cual es importante utilizar el Tablero Kanban, el mismo que fue realizado mediante el uso de la herramienta Trello siguiendo el flujo de trabajo de Kanban. A continuación, se presenta de manera gráfica la evolución del tablero de manera mensual.

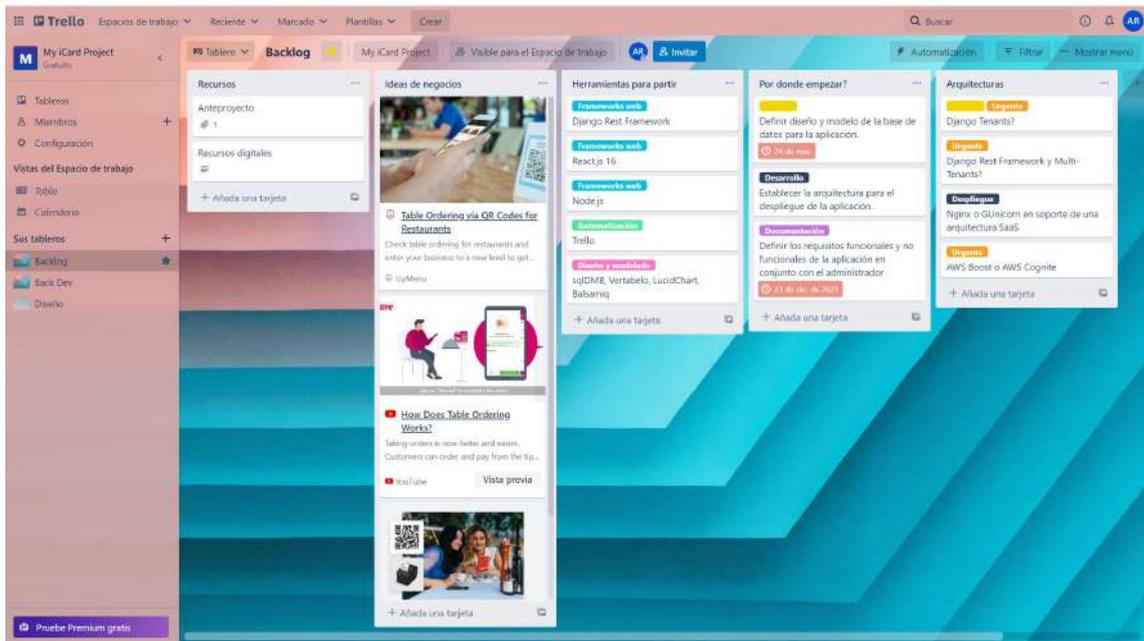


Ilustración 1: Tablero Kanban de “My iCard” al inicio del proyecto

Realizado por: Cuenca, A. y Llinín, F., 2022

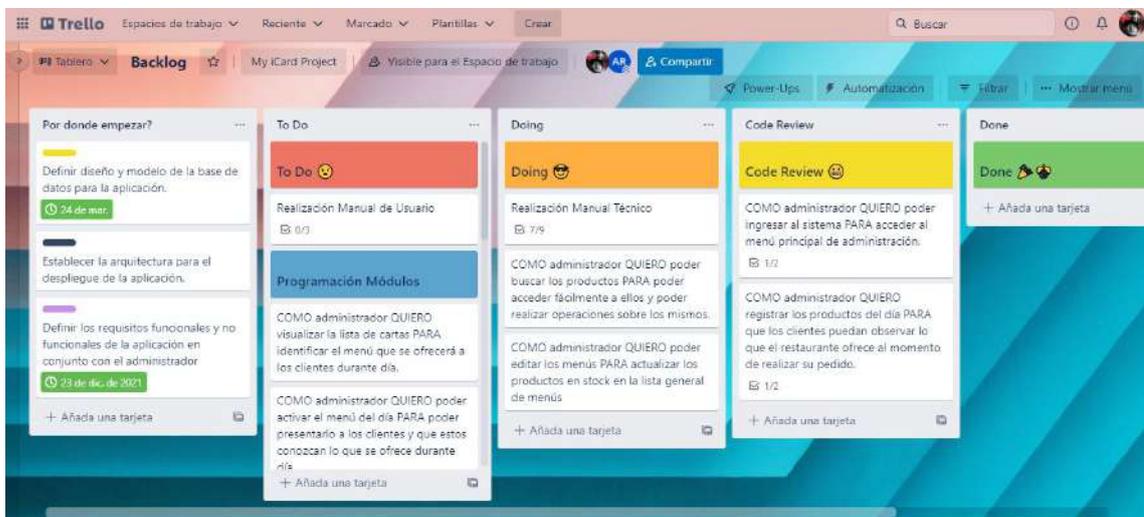


Ilustración 2: Tablero Kanban de “My iCard” al primer mes del proyecto

Realizado por: Cuenca, A. y Llinín, F., 2022

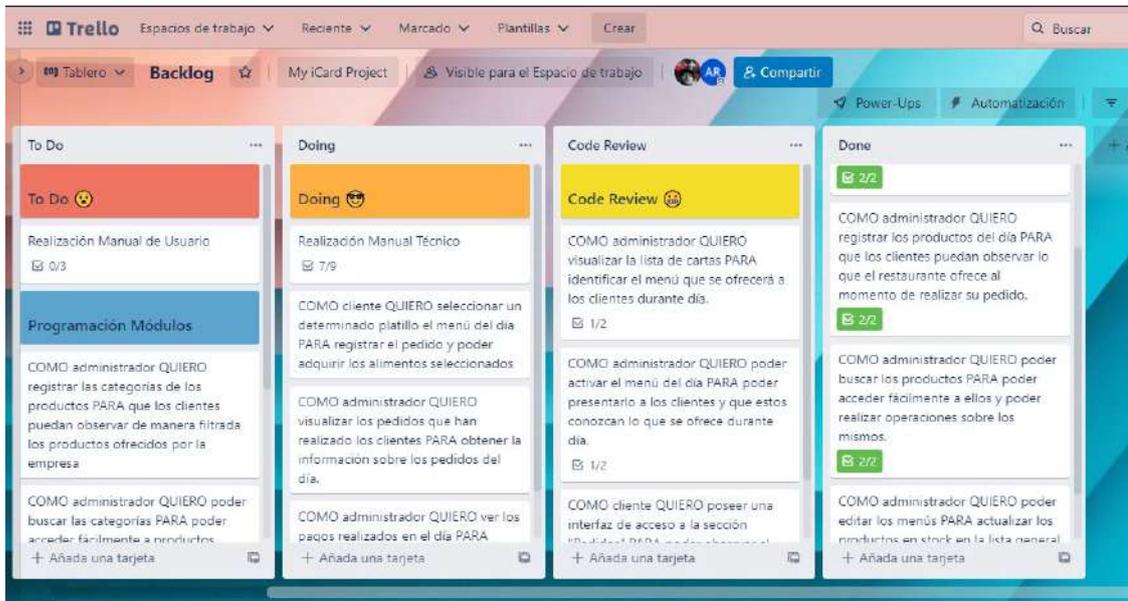


Ilustración 3: Tablero Kanban de “My iCard” al segundo mes del proyecto

Realizado por: Cuenca, A. y Llinín, F., 2022

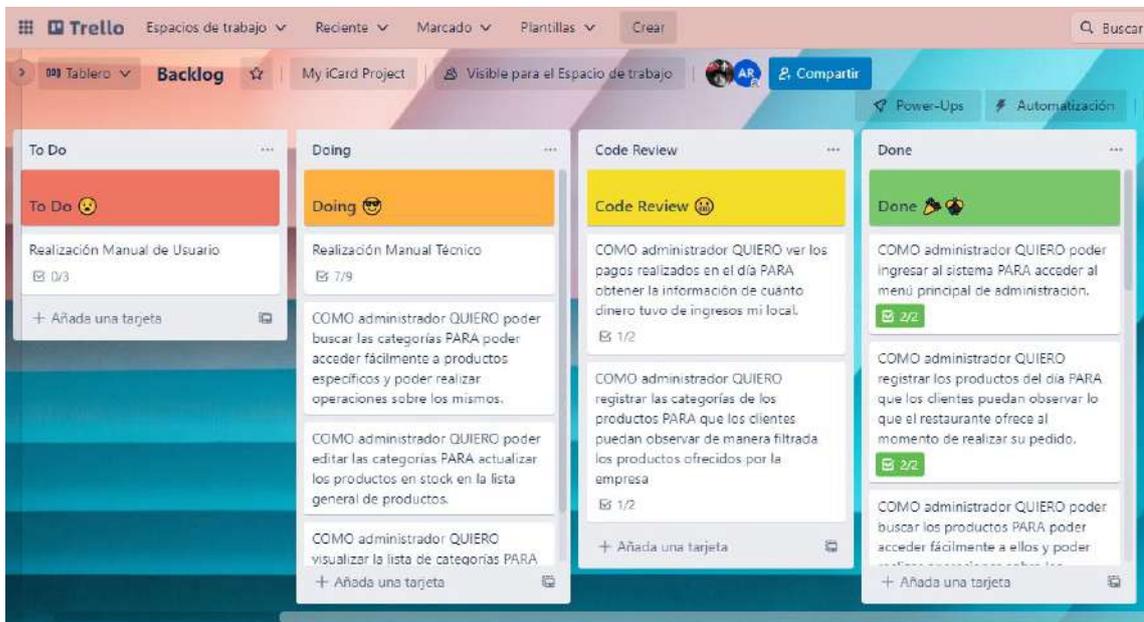


Ilustración 4: Tablero Kanban de “My iCard” al tercer mes del proyecto

Realizado por: Cuenca, A. y Llinín, F., 2022

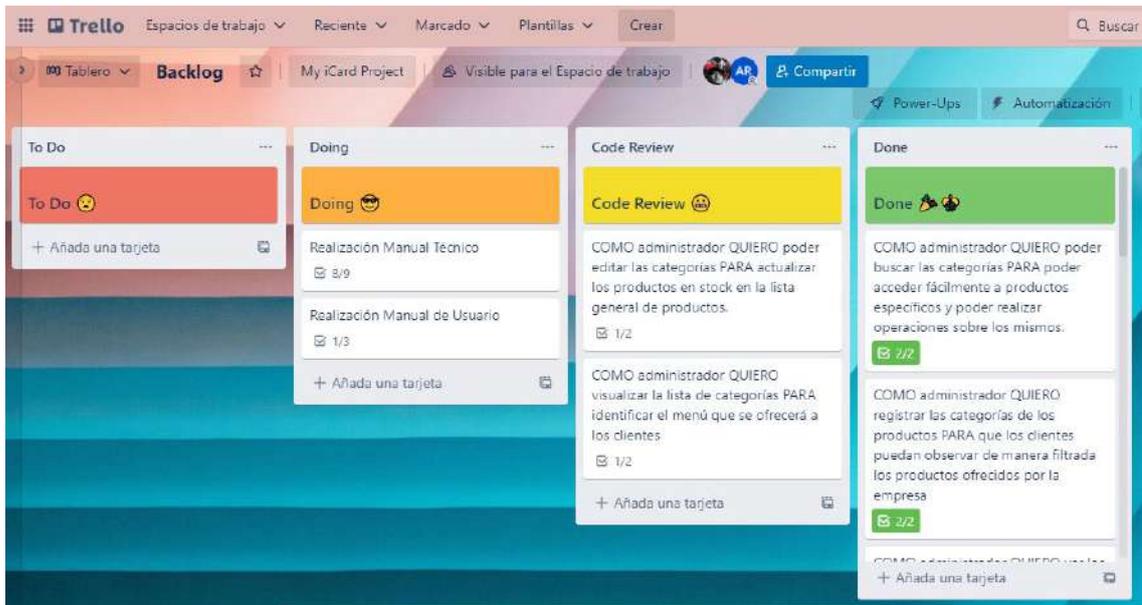


Ilustración 5: Tablero Kanban de “My iCard” al cuarto mes del proyecto

Realizado por: Cuenca, A. y Llinín, F., 2022

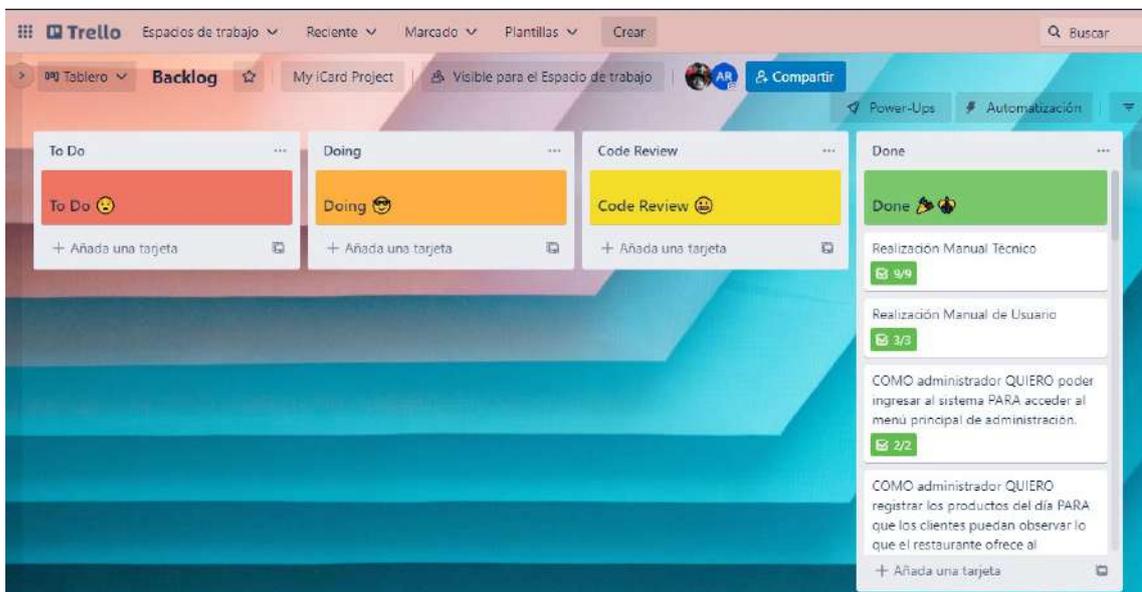


Ilustración 6: Tablero Kanban de “My iCard” al terminar el proyecto

Realizado por: Cuenca, A. y Llinín, F., 2022

ANEXO C: Documento validación del sistema “My iCard”



PIZZERIA D' BAGGIO

Riobamba, 2022-09-05

Srs.

Alex Cuenca, Francisco Llinin

ESTUDIANTES DE LA ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

Presentes.

De mi consideración:

Reciban un cordial saludo, el motivo de la presente es para validar el cumplimiento de lo pactado para el desarrollo del sistema web “My iCard” para mi establecimiento, con motivo de la realización de su Trabajo de Integración Curricular, indicando que el sistema funciona correctamente y logro dar solución a la problemática expuesta el primer día que empezó en la institución. El sistema es completamente funcional y el trabajo e interés que mostraron con mi empresa es digno de admirar, éxitos en las funciones que realice.

Particular que comunico para los fines pertinentes.

Atentamente;



Jorge Washington Flores Córdor

GERENTE DE PIZZERIA D' BAGGIO