

**Desarrollo de una aplicación, para el mejoramiento del servicio de la cafetería de la
Universitaria Agustiniana - U-Coffee.**

Daniel Fernando Torres Cubillos

Diana Mayerly García Macías

Universitaria Agustiniana
Facultad de ingeniería
Programa tecnología en desarrollo de software
Bogotá, D.C.
2020

**Desarrollo de una Aplicación APP, para el mejoramiento del servicio de la cafetería
de la Universitaria Agustiniana - U-Coffee.**

Daniel Fernando Torres Cubillos

Diana Mayerly García Macías

Director

Dúver René Acosta

Trabajo de grado para optar por título de Tecnólogo en desarrollo de software

Universitaria Agustiniana
Facultad de Ingeniería
Programa Tecnología en Desarrollo De Software
Bogotá, D.C.
2020

Resumen

El objetivo principal dentro del proceso inicial del proyecto era resolver el inconveniente visualizado en la cafetería de la Universitaria Agustiniiana a lo largo de los semestres cursados entre los años 2018 y 2020, en donde en compañía de varios estudiantes se concertó el problema de despacho de los pedidos en las horas más concurridas por estudiantes y administrativos. Como prueba inicial se tomó el tiempo promedio en la fila de la cafetería la cual podría variar de 15 a 20 minutos, adicionalmente, después de pasar el tiempo en dicha fila podría demorar la entrega de los alimentos de 10 a 15 minutos sin contar que en muchas ocasiones los artículos deseados podrían no estar disponibles. Se tomó como prueba inicial la creación de una aplicación que ayudara a resolver el obstáculo de la demora en la entrega de los pedidos, dentro de esto, se tomó como punto de partida los siguientes puntos: un sistema de pedidos, listado de productos disponibles, modificación de precios y artículos esto solo por parte del administrador de la cafetería, sistema de toma de pedidos que terminó traducido a un sistema ya conocido de aceptación de los pedidos por parte de la cafetería. Con la realización de estos procesos, se dispuso a la creación de la aplicación con los artículos más comunes vendidos por la Universitaria Agustiniiana, al finalizar esta parte se dispuso la creación de las pruebas expuestas en este documento para validar que todo funcionara de manera correcta, concretar cambios y sugerencias.

Palabras Clave: Pedidos, Productos, Artículos, Usuario, Cafetería.

Abstract

The main objective within the initial process of the project was to resolve the problem seen in the cafeteria of the Agustiniana University throughout the semesters between 2018 and 2020, where in the company of several students the problem of dispatching Orders in the busiest hours by students and administrators. As an initial test, the average time in the cafeteria line was taken, which could vary from 15 to 20 minutes, additionally, after spending the time in said line it could delay the delivery of food from 10 to 15 minutes without counting that in many times the desired items may not be available. The initial test was the creation of an application that would help solve the obstacle of the delay in the delivery of orders, within this, the following points were taken as a starting point: an order system, list of available products, modification of prices and items this only by the cafeteria administrator, an order-taking system that ended up being translated into an already known system of acceptance of orders by the cafeteria. With the completion of these processes, the creation of the application with the most common items sold by the Agustiniana University was prepared, at the end of this part the creation of the tests exposed in this document was arranged to validate that everything worked correctly, specify changes and suggestions.

Keywords: Orders, Products, Articles, User, Cafeteria.

Tabla de contenidos

Introducción.....	8
1. Título del proyecto de software	9
2. Planeación del proyecto.....	10
2.1. Objetivos del proyecto	10
2.1.1. Objetivo general.	10
2.1.2. Objetivos específicos.	10
2.2. Planteamiento del problema y/o necesidad.....	10
2.3. Alcance del proyecto	11
2.4. Metodología de desarrollo de documento.....	11
2.5. Metodología de desarrollo de software.....	18
2.5.1. Fase I exploración.	22
2.5.2. Fase II planeación.	22
2.5.3. Fase III diseño.	24
2.5.4. Fase IV codificación.	24
2.5.5. Fase V pruebas.	25
2.5.6. Fase VI muerte del proyecto.	25
3. Marco teórico y estado del arte.....	26
3.1. Marco teórico	26
3.1.1. Aplicación usuario.	26
3.1.2. Aplicación administrador.	26
3.1.3. Aplicación móvil.....	27
3.1.4. PHP.	27
3.1.5. IONIC.....	27
3.1.6. Base de datos.....	27
3.1.7. Página web.	27
3.2. Estado del arte.....	27
3.2.1. Rappi.	27
3.2.2. Domicilios.com.	28
3.2.3. PedidosYa.	28
3.2.4. ADN domicilios.	28
3.2.5. MacDonald's app.	28
3.2.6. El Corral app.	29
4. Especificación de requisitos de software (IEEE 830)	30
4.1. Perspectiva del producto	30

4.2. Funcionalidad del producto	30
4.2.1. Funciones aplicación de usuario.	30
4.2.2. Funciones aplicación de la cafetería.	30
4.3. Características de los usuarios	31
4.3.1. Usuario – cliente.	31
4.3.2. Usuario – cafetería.	31
4.4. Restricciones	31
4.5. Suposiciones y dependencias	31
4.6. Requisitos específicos	32
4.6.1. Actores/roles.	37
4.6.2. Requisitos funcionales.	38
4.6.3. Diagrama de casos de uso.	40
4.6.4. Especificaciones de los casos de uso.	41
4.7. Requisitos de rendimiento	44
4.8. Restricciones de diseño	44
4.9. Atributos del software del sistema	44
5. Diseño del software (ISO -12207-1)	46
5.1. Diseño de la arquitectura de software	46
5.1.1. Diagrama de contexto del sistema	46
5.1.2. Diagrama de contenedor.	47
5.1.3. Diagrama de componentes.	48
5.2. Diseño detallado del software	48
5.2.1. Diagrama de clases	48
5.2.2. Diagrama de paquetes.	49
5.2.3. Diagramas de despliegue	49
5.3. Diseño de la Interfaz	49
5.3.1. Interfaz gráfica de usuario	49
5.3.2. Interfaces de entrada	51
5.3.3. Interfaces de salida.	52
6. Implementación	53
6.1. Plataformas de desarrollo	53
6.2. Base de datos	53
6.3. Infraestructura de hardware y redes	53
6.3.1. Aplicación cafetería.	53
6.3.2. Aplicación usuario.	53

7. Pruebas del software	55
7.1. Pruebas del software	55
7.2. Pruebas de usabilidad.....	56
7.2.1. Estadística de pruebas.	58
Conclusiones.....	66
Referencias	67

Introducción

Siendo aprendices de las clases de la jornada de la tarde de la tecnología en desarrollo de software, pudimos validar por experiencia propia que la afluencia de personas dentro de la cafetería era muy alta en los instantes antes de los inicios de clases y últimos minutos de atención de la cafetería, esto nos llevó a idear como principal solución la creación de esta aplicación móvil para el usuario y la aplicación web para la gestión de la cafetería, la aplicación permite a todos los usuarios de la cafetería de la Universitaria Agustiniana ahorrar tiempo en las horas de alta afluencia de administrativos y estudiantes dentro de las instalaciones.

Esto ayudará siendo la tarea principal la creación del pedido antes de acercarse de manera presencial, para así tener el tiempo de ir a recogerlo con calma, esto evitará las filas, se optimizará el tiempo tanto para quienes usen los servicios de la cafetería como para quienes brindan dichos servicios.

Lo anterior, aparte de servir en lo ya indicado, podrá permitir a las personas encargadas de la cafetería a tener un mayor control de las ventas e inventarios.

Se pudo comprobar teniendo como guía a varias aplicaciones de entrega de comida, que al tener el tiempo “medido” o saber cuánto tardaremos en tener nuestros alimentos podemos optimizar los instantes póstumos a reclamar los alimentos y así evitar el estrés de tener que hacer largas filas o perder instantes de clases.

1. Título del proyecto de software

Desarrollo de una aplicación móvil, para el mejoramiento del servicio de la cafetería de la Universitaria Agustiniana - U-Coffee.

2. Planeación del proyecto

2.1. Objetivos del proyecto

2.1.1. Objetivo general.

Desarrollar una aplicación móvil para la automatización del sistema de pedidos en la cafetería de la Universitaria Agustiniana utilizada por los estudiantes, profesores y directivos, mejorando el tiempo de servicio de entrega de los alimentos, además, de brindar la opción de generar opiniones sobre los alimentos entregados y el servicio brindado.

2.1.2. Objetivos específicos.

1. Brindar a los estudiantes de Universitaria Agustiniana la opción de tener otros medios para solicitar sus alimentos dentro de las instalaciones de la universidad.
2. Optimizar el uso de la metodología elegida para el desarrollo del proyecto: aplicación para la cafetería de la Universitaria Agustiniana, dando un uso e implementación completo desde el inicio y así lograr las metas propuestas para el mismo.
3. Reducir el tiempo de espera por parte de los usuarios de la cafetería de la Universitaria Agustiniana en las horas de alta afluencia de estudiantes y administrativos.
4. Definir y delimitar la funcionalidad de la aplicación móvil de entrega de alimentos en la cafetería de la Universitaria Agustiniana por medio de las opiniones de los miembros de la comunidad estudiantil.

2.2. Planteamiento del problema y/o necesidad

La aplicación U-Coffee como modelo inicial, busca mediante un proceso de realización de ordenes brindar el correcto funcionamiento de entregas de los alimentos de la cafetería de la Universitaria Agustiniana, centrado en las horas de alta afluencia de usuarios.

Los procesos de la misma se ven retrasados en las horas “pico” en donde a lo largo de nuestra estadía dentro de las instalaciones de la universitaria pudimos visualizar son en las horas comprendidas entre las 18:00 y 18:30, como primera jornada y posterior a ello, de 19:45 hasta las 20:30, la mayor cantidad de estudiantes, profesores y administrativos, en estas horas se debe a que en estas franjas corresponden los inicios de clase de la jornada nocturna y el cambio de clases, allí, podemos ver como todos, sin excepción, realizan largas filas para comprar los artículos que ofrece la cafetería, adicionalmente, al no tener un correcto control, la mayoría de las veces las personas ubicadas al final de la fila tienen muy pocas opciones para pedir.

Se logró identificar que no se cuenta con proceso sistematizado para entrega de alimentos ni control de inventarios por este medio tampoco, además, al hacer una búsqueda específica se logra encontrar que ninguna universidad a nivel nacional cuenta con una aplicación de entregas

para la misma universidad, sin embargo, si se cuentan con modelos de negocio de entrega de comidas de otras empresas, por lo que se parte desde esta perspectiva en este proyecto.

2.3. Alcance del proyecto

La aplicación se implementará en la Universitaria Agustiniiana sede Tagaste y posteriormente se podrá descargar en la sede Suba, la misma podrá ser utilizada por usuarios Android y IOS. Se espera que la aplicación sea descargada por toda la comunidad de la Agustiniiana.

El proyecto espera disminuir los tiempos de espera dentro de las clases e inicios de jornada, tanto para estudiantes, administrativos y visitantes de la Universitaria Agustiniiana, esto a su vez ayudará a la cafetería a tener control sobre los procesos que se realizan; las ventas, inventario y horas en las que se pedirá un artículo más que otro.

Esto se logrará a través de dos formatos, el primero siendo la aplicación de usuario que se descargará por medio de una versión descargable y la aplicación de la cafetería, al ser un servicio web el administrador creará una cuenta y allí se vinculará a la aplicación de usuario, de esta manera empezará a aceptar los pedidos, cambiar o editar los artículos ya disponibles o agregar nuevos si así lo desea.

2.4. Metodología de desarrollo de documento

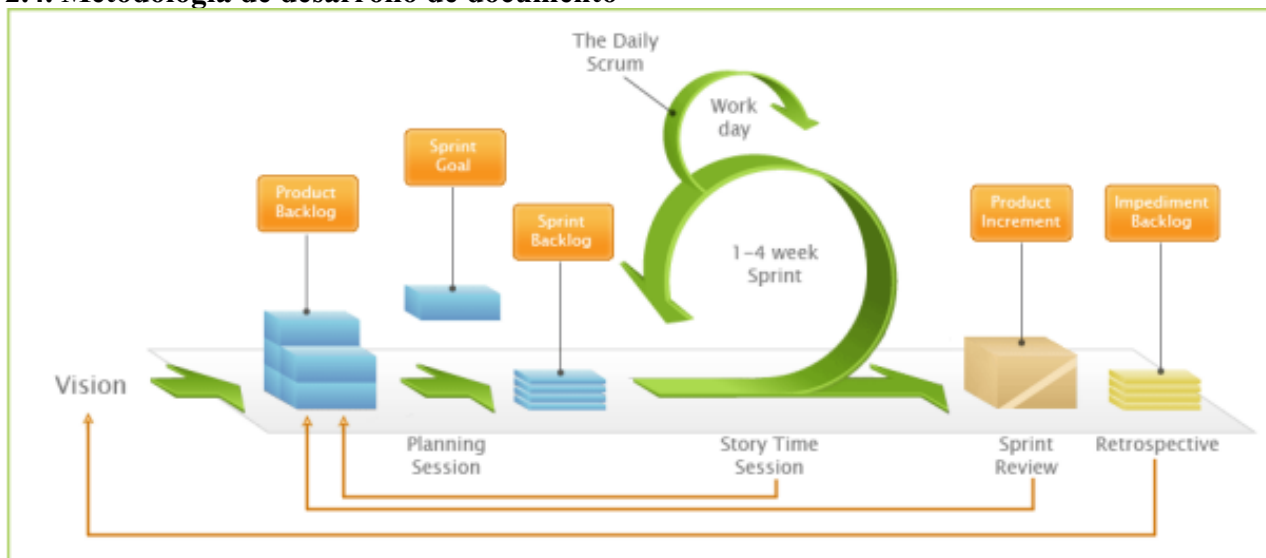


Figura 1. Metodología Scrum. Manuel Trigas Gallego. (s.f.).

Para el desarrollo de este proyecto se establecen criterios tanto para la investigación como para la realización del software, además de las opiniones de profesores, estudiantes, administrativos y miembros de la cafetería de la Universitaria Agustiniiana, lo anterior, se busca

hacerlo mediante diálogos establecidos con alguno de ellos o envío de formas para escuchar las opiniones.

La metodología que se ha elegido para aplicar al proyecto escrito será la Scrum, ya que propone procesos de mejora continua, esto debido a que todos los miembros del equipo tienen conocimiento de lo que ocurre en todas las fases del proyecto.

Un resumen más claro podría ser el brindado por Trigas (2012): “Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán "Sprints".” (p. 33).

Debido a que dentro de la metodología elegida los ciclos son temporales y se tendrán iteraciones de dos semanas, se podrá tener un proceso general de cómo va el proyecto y todos los miembros del grupo podrán opinar sobre la realización. Para tener un concepto claro sobre los procesos que se tomarán para la realización del proyecto tomamos los indicados por Trigas (2012):

Concepto: Se define de forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.

Especulación: En esta fase se hacen disposiciones con la información obtenida y se establecen los límites que marcarán el desarrollo del producto, tales como costes y agendas. Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas su impacto en el entorno.

Esta fase se repite en cada iteración y consiste en rasgos generales en:

Desarrollar y revisar los requisitos generales.

Mantener la lista de las funcionalidades que se esperan.

Plan de entrega, se establecen las fechas de las versiones, hitos e iteraciones.

3. Exploración: Se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.

4. Revisión: El equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.

5. Cierre: Se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá haciendo cambios, denominados “Mantenimiento”, que hará que el producto final se acerque al producto final deseado. (p.33).

El desarrollo de la aplicación se hará a través de las herramientas tecnológicas móviles híbridos, esto para permitir que tanto los usuarios Android como IOS, que son los más usados por la comunidad Agustiniense, puedan usarla, adicional, un Sistema Gestor de Bases de Datos para el almacenamiento de la información.

La interfaz gráfica con la que contará la aplicación será intuitiva, simple, interactiva e inspirada en la Universitaria Agustiniense, tendrá disponibilidad en los horarios de la cafetería.

La aplicación se desarrollará en el framework IONIC utilizando los lenguajes CSS3 (Cascading Style Sheets), JavaScript y html5 (HyperText Markup Language), esto nos ayudará a que la elaboración sea sencilla y eficiente.

Como anteriormente se expuso, IONIC permite que el producto final pueda ser exportado a los sistemas operativos, cómo podemos verlo directamente expuesto en la página de IOS (Sistema operativo propio de Apple Inc):

iOS es un sistema operativo orientado a equipos de escritorio que ha sido reinventado para dispositivos móviles. Ya que está basado en el sistema operativo informático más avanzado del mundo, OS X, iOS presenta un rápido desempeño y una estabilidad muy sólida. Ahorra energía y permite una asombrosa duración de la batería. iOS incluso garantiza que el desempeño y la duración de la batería no sufran cuando se está usando varias APPS al mismo tiempo (Correa, 2013, p. 43).

De igual manera Android (Sistema operativo propiedad de Google) que podemos visualizar:

Es una stack de software para dispositivos móviles que incluye un sistema operativo, middleware y aplicaciones base. Los desarrolladores pueden crear aplicaciones para la plataforma usando el SDK de android. Las solicitudes se han escrito utilizando el lenguaje de programación Java y se ejecutan en Dalvik, una máquina virtual personalizada que se ejecuta en la parte superior de un núcleo de Linux. (Cruz, Gopar, López, Moya, 2008, p.2).

En caso de ser necesario adaptarse a un nuevo sistema operativo podrá ser ajustado mediante una nueva actualización, la aplicación necesitará de acceso a internet para poder ser usada de manera óptima.

Para tener un control final de las fases de la metodología se realiza la descripción de los procesos ya inmersos:

Concepto: En el cual se brindaron las ideas para la creación del proyecto aquí planteado, se dividieron las tareas, en donde se hizo la identificación de los procesos que se deseaban tener para hacer la entrega del proyecto correctamente en su última fase.

Dentro de este se dividió la información y se pasó al proceso de Gantt en donde se dispusieron las semanas que teníamos para culminar el proyecto, esto dentro de los plazos brindados en la Universitaria Agustiniiana, de igual forma, se tomó en cuenta el tiempo de las revisiones de cada una de las fases en el transcurso de nuestro último semestre dentro de la Universitaria.

La división de las tareas las veremos en mayor parte de la siguiente manera:

Daniel Fernando Torres Cubillos: Programar y tester.

Diana Mayerly García Macías: Desarrollo conceptual y tester.

Especulación: En esta fase se realizaron las disposiciones con la información que tuvimos de búsqueda y se establecieron los cronogramas y costes que podemos visualizar en el diagrama de Gantt adjunto, en esta fase se repiten todas las iteraciones por lo que se especifica cuantas se han tenido y que se ha revisado en cada una:

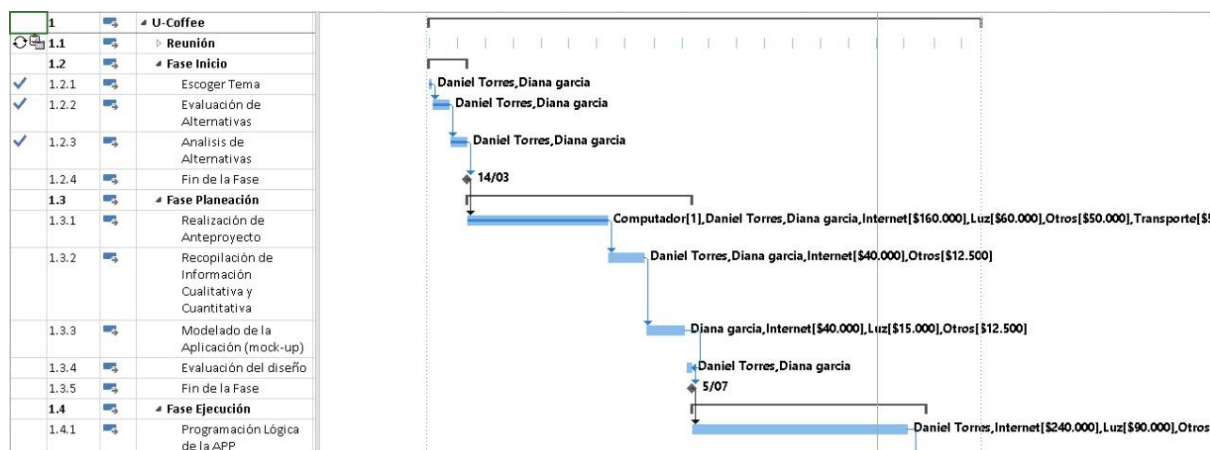


Figura 2. Diagrama de Gantt primera sección. Autoría propia.



Figura 3. Diagrama de Gantt segunda sección. Autoría propia.

Se llega al acuerdo de que se tendrán un total de 3 iteraciones a lo largo de los meses faltantes para la finalización del proyecto, en total una por mes, en donde se brindarán los aportes realizados, correcciones, sugerencias y revisión de la agenda tomada del diagrama realizado en la primera fase.

Se dispone el tiempo de realización del proyecto a un total de 12 horas semanales, un total de dos horas diarias de lunes a viernes, para tener de igual forma tiempo para las demás actividades de las que disponemos durante este semestre.

Iteración I:

Revisión de los objetivos.

Replanteamiento del nombre del proyecto.

Se establecen iteraciones 1 por mes.

Iteración II:

Revisión de desarrollo de la aplicación de usuario.

Revisión de casos y Mockups para la aplicación de la cafetería.

Se brindan los siguientes wireframes para el seguimiento del código.

Se realiza la creación del logro que llevará la aplicación.

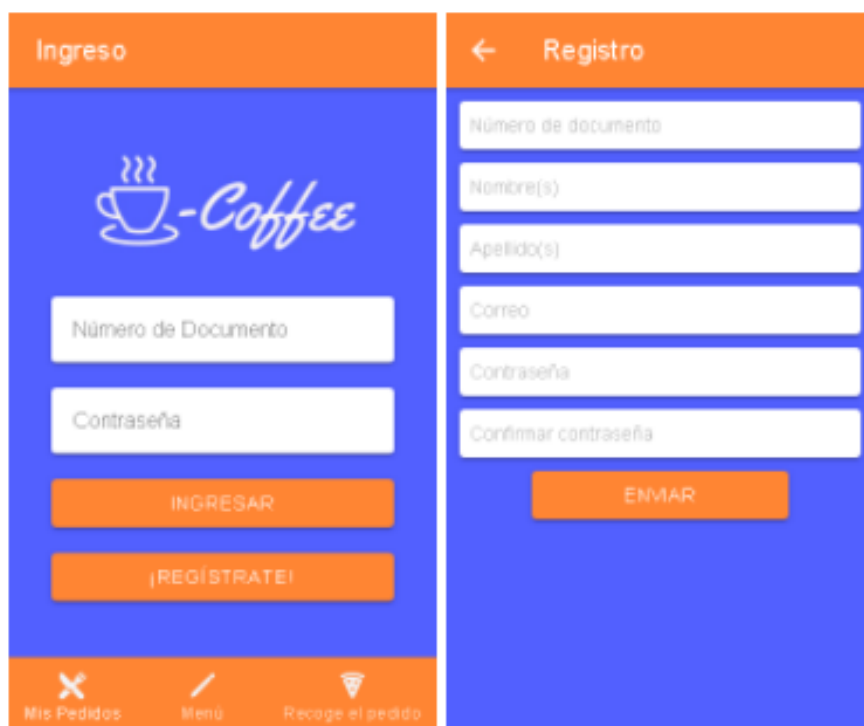


Figura 4. Wireframe ingreso y registro aplicación de usuario. Autoría propia.



Figura 5. Wireframe desarrollo de pedidos aplicación de usuario. Autoría propia.

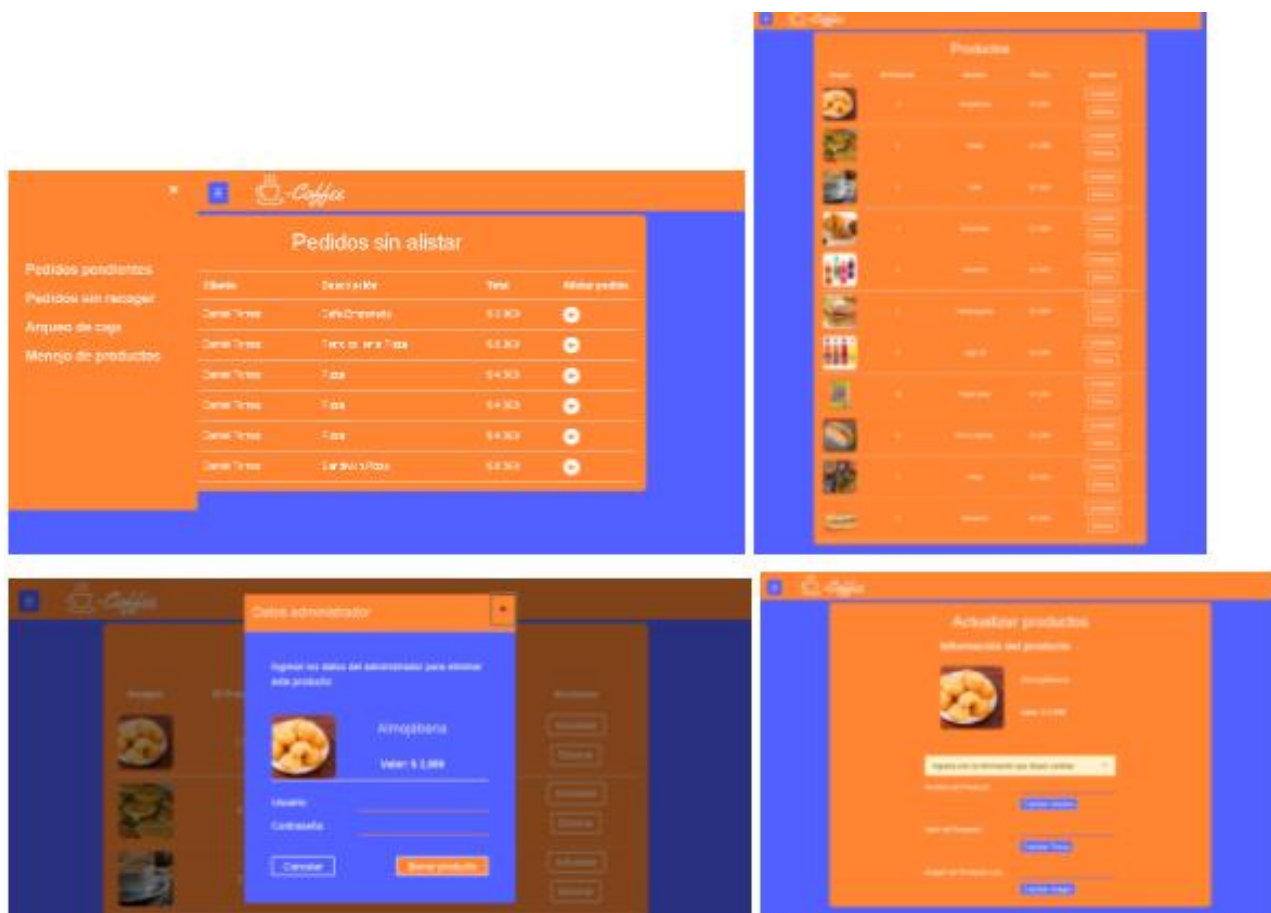


Figura 6. Wireframe edición o manipulación de artículo servicio web de la cafetería. Autoría propia.



Figura 7. Wireframe desarrollo de pedidos servicio web de la cafetería. Autoría propia.



Figura 8. Logo del proyecto U-Coffee. Autoría propia.

Iteración III:

Revisión de los avances de la aplicación y funcionamiento de la base de datos dentro del desarrollo general de la aplicación, ya que como fue previsto desde el inicio mediante el proyecto avanzaba se encontraron escenarios o cambios no previstos, durante las iteraciones se buscaba impactar en la menor medida el código ya realizado, sin embargo, cumpliendo con los cambios y sugerencias pensados.

Exploración: Se inicia creación de la aplicación en código, se continua con las iteraciones cada mes (30) días para ir visualizando errores y procesos de mejora de esta, todos las ideas y correcciones dadas mediante las iteraciones se tienen en cuenta para cambios en esta fase.

Debido a que en el proyecto se dieron ideas en las iteraciones planteadas se fue actualizando el código, interfaz y por ende el proyecto en general, las ideas brindadas se pasan a comité con los miembros del grupo para posterior a ello, ser aprobadas e implementadas.

Revisión: Dentro de la revisión se integró la visualización con la última entrega de la clase de trabajo de grado en donde se mostraron todos los avances del proyecto.

Cierre: Se da entre el día 13 de noviembre con todas las correcciones dadas.

2.5. Metodología de desarrollo de software



Figura 9. Metodología XP. s.n. (s.f.).

En relación con la implementación de desarrollo de software y del documento se validó el proceso de creación de dos aspectos, metodología de desarrollo del documento y del software de manera general, esto con el fin de dejar de manera específica la composición del producto.

Para este proceso se toma en consideración la metodología de desarrollo de software eXtreme Programming (XP), se decide esta ya que al ser una clasificación ágil permite tener consideraciones dentro del personal implicado en el proyecto desde el inicio al final del proceso y la entrega del proyecto.

Dentro de la metodología se promueve el trabajo en equipo en todas las fases y esto permite, al ser de igual manera, un proceso ágil que existe un proceso flexible en donde podemos manifestar cronogramas de actividades flexibles y claros.

Los roles del proyecto las podemos visualizar como dice Letelier, (2006):

Programador:

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

Ciente:

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

Encargado de pruebas (*Tester*):

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (*Tracker*):

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Entrenador (*Coach*):

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor:

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (*Big boss*):

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación. (p. 1).

De igual manera podemos mencionar las seis fases según Carrasco, Ocampo, Ulloa, Azcona (2019):

Fase I: Exploración Para empezar, es necesario definir los roles de las personas involucradas en el proyecto, como lo pueden ser: Product Owner, Development Team y Scrum Master. En esta primera fase se ha buscado realizar una reunión con el Product Owner y con todos los interesados del negocio, el cual realiza la función de intermediario entre los interesados y el equipo de desarrollo para recolectar las funcionalidades que la empresa necesita, con la finalidad de que se puedan plantear las

historias de usuario que son de interés para la entrega del producto. Estas historias son los ítems del Product Backlog.

Fase II: Planeación A partir de esta fase se da inicio con el primer sprint. Para empezar, se eligen las historias de usuario de acuerdo a la prioridad que se tienen en el Product Backlog, cuyos puntos de estimación totales no superen la velocidad de desarrollo del equipo. Una vez que se han establecido las prioridades de las historias de usuario, el Development Team procede a estimar la complejidad de cada una de ellas haciendo uso de los puntos de historia, métrica utilizada en las metodologías de desarrollo ágil. Esto último se logra asignando un valor estimado a la complejidad (haciendo uso de la serie Fibonacci) de acuerdo a la experiencia de los desarrolladores. Por consiguiente, se crea el Sprint Backlog en base a las historias de usuario elegidas, dividiéndolas en tareas de ingeniería para llevar un mejor control del trabajo pendiente, la duración de cada sprint que será de cuatro semanas. Se recomienda que en esta fase se apoye en la técnica mockups, para realizar una fertilización cruzada, escuchar las necesidades del Product Owner (voz y mente del cliente) y se define los escenarios de pruebas.

Gestión de Tareas: Con el objetivo de llevar un control preciso de las historias de usuario y sus respectivas tareas de ingeniería, se ha empleado un software para la gestión de tareas llamado Trello. Este software hace uso del sistema Kanban (sistema de tarjetas) en una interfaz web que ha facilitado la gestión del desarrollo del proyecto de acuerdo a la organización que propone Scrum, donde se controla el trabajo pendiente y se organiza el trabajo terminado en las diferentes columnas: en espera, en desarrollo, por validar y completo. A manera de ejemplo, se presenta el enlace al tablero Trello que se ha realizado con anterioridad: <https://trello.com/b/lZ7ApRPF>.

Gestión de Versiones: Para facilitar los procesos de respaldo del sistema y gestionar las versiones que se vayan desarrollando, se optó por utilizar Github, una excelente plataforma enfocada al desarrollo colaborativo de software. Github utiliza Git, un sistema de control de versiones que permite alojar repositorios en la web, permitiendo crear una copia local del código y respaldarla para su posterior revisión, uso y gestión; además, permite volver a versiones anteriores del sistema en caso de que se haya presentado algún error.

Fase III: Diseño La fase de diseño se enfoca en el uso de las tarjetas CRC (Clase-Responsabilidad-Colaborador). Estas tarjetas ayudan a identificar y organizar fácilmente las clases más relevantes para los requerimientos de un sistema; además, se dividen en tres secciones: en la parte superior se encuentra el nombre de la clase, en la parte izquierda se enlistan todas las acciones que la clase pueda realizar (responsabilidades) y en la parte derecha, otras clases que interactúen con la actual (colaboradores), ayudándola a llevar a cabo sus responsabilidades. Fase IV: Codificación Aquí es donde se desarrollan las funcionalidades del sistema, escribiendo el código fuente necesario para cumplir con lo detallado en las tareas de ingeniería. Cabe mencionar que diariamente se ha realizado

el evento Daily Scrum, reunión que ayudó a verificar el trabajo realizado el día anterior, visualizar las tareas que se desarrollarán a lo largo del día e identificar los riesgos que se han presentado.

Fase V: Pruebas Es necesario aclarar que, a lo largo de la fase anterior, se realizaron pruebas unitarias automatizadas con el objetivo de verificar que el desarrollo continuo no afecta a otras funcionalidades. Retomando esta última fase del sprint, se utilizó el evento Sprint Review, reunión en compañía del Product Owner donde se realizó la demostración del producto mínimo viable (incremento), explicando las funcionalidades desarrolladas; además, esta reunión se apoyó en las pruebas de aceptación, las cuales sirven como constancia de que el incremento fue desarrollado con éxito.

Una vez terminado el desarrollo del sprint, como resultado del control diario de las tareas de ingeniería (Daily Scrum) se obtendrá el gráfico de avance, también conocido como Burndown Chart, el cual permite apreciar las tareas pendientes. Finalmente, también se realizó el evento Sprint Retrospective, reunión realizada luego de la revisión del sprint, cuyo objetivo es el de analizar detenidamente el trabajo realizado a lo largo del sprint para identificar las diferentes maneras de cómo mejorar la forma de trabajo.

A partir de esta fase se da por concluido el sprint correspondiente. Es necesario volver a la fase de Planeación (Fase II) para continuar con el siguiente sprint y así sucesivamente. Una vez que todos los sprints hayan sido culminados con éxito, se procede a la siguiente fase.

Fase VI: Muerte del Proyecto Una vez que el Product Backlog no tiene más historias de usuario y todas las funcionalidades requeridas fueron desarrolladas satisfactoriamente, se da por terminado el ciclo de vida de desarrollo del software. En esta última fase se realiza la documentación final que corrobora la entrega y aceptación del sistema por parte del Product Owner, como por ejemplo una Acta de Entrega-Recepción del Proyecto. (p. 112 - 115).

Llevando la información al desarrollo del producto se traduce de la siguiente manera brindando la clasificación de los roles:

Programador: Dentro de este rol se encarga Daniel Torres, debido al proceso de creación y codificación de la aplicación móvil y web.

Cliente: La figura del cliente se llevará a cabo inicialmente por Daniel Torres y Diana García que harán las veces de este rol con el fin de aclarar procesos y actualizaciones.

Encargado de pruebas: Este rol será encargado por Diana García, quien realizará las pruebas para todos los usuarios (cliente, desarrollador y cafetería), de igual manera, se encarga de la distribución de estas.

Encargado de seguimiento: Estas serán desarrolladas por Diana García, quien será la encargada de informar en que proceso se está llevando tanto el proyecto de manera escrita como funcional.

Entrenador: De este proceso será encargado Daniel Torres, ya que dispondrá la información de los procesos de XP con el proyecto general.

Gestor: El gestor general será Diana García que recibirá la información de XP para llevarla al proyecto general y entregada al final en la muerte del proyecto.

2.5.1. Fase I exploración.

Se realiza la primera visualización del proyecto, en donde se validan el objetivo de este, partiendo de este punto se logra dividir los requerimientos de cada miembro del producto final, se establecen tiempos y las iteraciones que están establecidas en el desarrollo de la metodología del documento Scrum.

2.5.2. Fase II planeación.

Dentro del proceso de planeación del proyecto, se realiza la división del proyecto de la siguiente manera:

Historias de Usuario:

Tabla 1.

Historias de usuario I

Nombre	Validación de roles
Descripción	En donde se visualizarán la división de los roles de los procesos, se llega al acuerdo de tener una aplicación móvil de usuario y una aplicación web para la cafetería.

Nota. Autoría propia. Tabla de validación de roles I.

Tabla 2.

Historias de usuario II

Nombre	Creación de usuarios
Descripción	En donde se validarán las creaciones de los usuarios de ambos roles y que estos queden guardados en la base de datos

Nota. Autoría propia. Tabla de validación de roles II.

Tabla 3.

Historias de usuario II

Nombre	Creación de ordenes
Descripción	Se debe poder crear la orden y posterior a ella aceptarla para que el proceso sea culminado

Nota. Autoría propia. Tabla de validación de roles III.

Tabla 4.

Historias de usuario IV

Nombre	Cierre de ordenes
Descripción	Se debe poder cerrar la orden y que la misma no siga activa en ninguno de los perfiles

Nota. Autoría propia. Tabla de validación de roles IV.

Tabla 5.

Historias de usuario V

Nombre	Arqueo de caja
Descripción	El sistema administrativo de la cafetería deberá poder realizar un arqueo final de caja

Nota. Autoría propia. Tabla de validación de roles V.

Velocidad del proyecto: Dentro de las iteraciones elegidas para el proceso de desarrollo de la metodología de software se eligieron 1 cada dos semanas.

Iteraciones: Se establecen 3 iteraciones una cada dos semanas en donde establecerán de la siguiente manera con los procesos y las actividades de cada una:

En la primera iteración se establecerán los diseños iniciales de proyección de los dos aplicativos, se establecen los lenguajes a trabajar para cada una de las aplicaciones y el plan de entregas de los procesos de las aplicaciones.

En la segunda iteración se realizan las revisiones correspondientes de los avances, se establecen los objetivos y puntos de las siguientes iteraciones.

En la tercera iteración se establecen los procesos de las pruebas finales después de la finalización de las iteraciones anteriores, esto se llevará a cabo entre las dos partes de los responsables del proyecto.

Plan de entregas: Se realiza en la primera iteración la creación del cronograma de planeación, en las mismas se estimó un proceso de trabajo de 2 horas diarias, todos los procesos quedan especificados en el diagrama de Gantt:

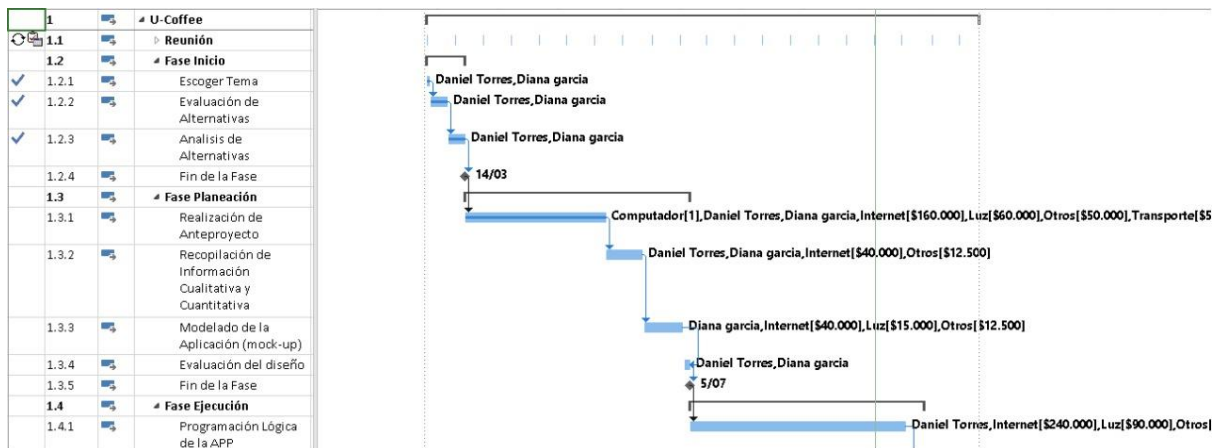


Figura 10. Diagrama de Gantt 1. Autoría propia.



Figura 11. Diagrama Gantt 2. Autoría propia.

2.5.3. Fase III diseño.

Dentro de las proyecciones importantes revisadas en el diseño de la metodología de software XP se validan:

Simplicidad: En donde se trabajó en que la aplicación tanto web como móvil para que la misma se mantuviese intuitiva, ágil, y básica.

Refactorización: Dentro de las cuales se validaron los procesos que no se tuvieron contemplados en un inicio, sin embargo, al tener consideraciones se decidieron implementar para la mejora de la aplicación, para que este desarrollo se hiciera de la manera correcta se tuvieron contemplaciones para no afectar todo el diseño o código de manera general.

2.5.4. Fase IV codificación.

Para que la codificación se llevara a cabo de la manera en que la metodología de software escogida sugiere, se dividieron los procesos en:

Presencia del cliente: Durante todo el proceso de la creación de las aplicaciones se mantuvo contacto directo con el cliente y desarrollador para validar todos los cambios y admitir las sugerencias de las dos partes.

Estándares y propiedad colectiva del código: Para mantener el código de forma segura, se mantuvieron algunos comentarios dentro de mismo para el entendimiento de las líneas solo para los desarrolladores, sin embargo, en caso de que otra persona tuviese que unirse con los comentarios es fácil de entender.

Producción de código hecha en parejas: El código fue realizado por el programador Daniel Torres, sin embargo, el otro compañero participó en el diseño de las aplicaciones.

Codificar de las pruebas: Se realizan pruebas durante el desarrollo de las aplicaciones, así, de esta forma se pueden validar los aspectos de correcciones de código o diseño.

2.5.5. Fase V pruebas.

Dentro de la realización de las pruebas de usuario se realizó la contemplación de las pruebas unitarias, estas se dividieron en dos:

Pruebas de desarrolladores – creadores: Dentro de estas pruebas se toman en consideración las realizadas desde el inicio de la creación del proyecto en cuanto a código, durante las iteraciones se mostraban los avances y se hacen pruebas con dichos avances y de esta manera se crean las sugerencias.

Pruebas de desarrolladores - compañeros: Las pruebas de los desarrolladores invitados se realizaron con un formato y la última versión de la aplicación, aquí se validaron dos aspectos importantes; diseño y uso general, los resultados de las mismas están al finalizar el proyecto.

2.5.6. Fase VI muerte del proyecto.

Al dar por finalizado el proyecto y las iteraciones el cliente no presenta más historias o sugerencias con la misma. Se generan como es de esperar la documentación final y no se realizan más cambios. La finalización del proyecto también se da, cuando no se suponen más inconvenientes.

3. Marco teórico y estado del arte

3.1. Marco teórico

3.1.1. Aplicación usuario.

Al no tener predecesores en este proyecto, la idea plasmada es: brindar inicialmente la aplicación a la comunidad de la Universitaria Agustiniense, en ella encontrarán el listado de los alimentos ofrecidos por la cafetería, en caso de no tener alguno disponible, se indicará mediante un mensaje o directamente no estará disponible en ese momento, al momento de hacer el pedido, el usuario podrá elegir el método de pago, allí le mostrará cuánto puede tardar su pedido en ser preparado y entregado, después le indicará que debe acercarse a la cafetería, hacer el pago y retirar sus alimentos.

Al finalizar su pedido, tendrá una opción de “Comentarios del Pedido” y “Comentarios de Servicio” allí los usuarios de la cafetería podrán brindar su opinión a la cafetería.

Los módulos propuestos para la etapa inicial de este proyecto son:

Módulo de diseño de entrega.

Módulo de visualización de inventarios.

Módulo de asignación de tareas.

Módulo de diseño general de la aplicación.

3.1.2. Aplicación administrador.

Desde este perfil se podrán realizar las labores de aceptar los pedidos, se espera que el 100% de los mismos sean aceptados y entregados, al ser la segunda parte involucrada en este caso para que el círculo de la entrega esté completo se desarrollará de la siguiente manera: al momento de envío del pedido por parte del usuario los miembros de la cafetería lo podrán aceptar, al hacerlo, les aparecerá un desglose con los artículos, ellos los preparan, calentarán o servirán, esto dependiendo de la orden.

Al finalizar el pedido, indicarán que ya está listo para ser recogido en la cafetería, el usuario se acercará, pagará y se finalizará esa entrega.

Adicional al proceso principal, la cafetería tendrá la opción de validar los comentarios tanto de los alimentos como de los miembros del equipo de Staff de la cafetería, esto ayudará con el mejoramiento continuo del servicio y conocimiento de los alimentos que más se consumen o que casi no se piden por la comunidad universitaria y así aumentar o disminuir el inventario.

3.1.3. Aplicación móvil.

Las aplicaciones móviles son nuestro diario vivir, existe una para cada necesidad, criterio, gusto y necesidad y la podemos definir según el diccionario electrónico EcuRed (2018a):

Una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros. Existen aplicaciones móviles gratuitas u otras de pago, donde en promedio el 20-30% del costo de la aplicación se destina al distribuidor y el resto es para el desarrollador.

3.1.4. PHP.

Se define según Hypertext Preprocessor (PHP) (s.f.): “Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML”.

3.1.5. IONIC.

Se define según Ionic framework (s.f.):

Es un kit de herramientas de interfaz de usuario de código abierto centrado en la creación de aplicaciones móviles de alta calidad para iOS nativo, Android nativo, ¡y la web! Desde cero, está construido con HTML, CSS y JavaScript, por lo que los desarrolladores web deben sentirse como en casa con nuestros componentes. Nuestros componentes permiten a los desarrolladores crear experiencias nativas, todo mientras utilizan la tecnología web.

3.1.6. Base de datos.

Se define por el diccionario electrónico EcuRed (2018b):

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una Biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

3.1.7. Página web.

Se define por el diccionario electrónico EcuRed (2018c):

Una página web no es simplemente una aplicación del diseño convencional, ya que, además, hay que considerar una serie de elementos que van a condicionar el desarrollo exitoso del proceso de diseño tales como navegabilidad, interactividad, usabilidad, arquitectura de la información y la interacción de medios como el audio, texto, imagen y vídeo.

3.2. Estado del arte

3.2.1. Rappi.

Es una multinacional de origen colombiano que tiene como servicio principal la entrega de productos, la aplicación permite a sus usuarios ordenar desde mercado hasta productos preparados, medicamentos, productos para mascotas, artículos para hacer ejercicio entre otros.

Fundadores: Felipe Villamarín, Sebastián Mejía, Simón Borrero

Lugar: sede principal Bogotá D.C, Colombia.

Fecha: 2015.

3.2.2. Domicilios.com.

Surgió como un plan piloto entre dos estudiantes de la universidad de los Andes, los cuales tenían la idea de pedir comida en línea de una manera fácil y segura, actualmente es el único emprendimiento tecnológico colombiano que cotiza en la bolsa de Frankfurt.

Fundadores: José Guillermo Calderón y Miguel McAllister.

Lugar: sede principal Bogotá D.C, Colombia

Fecha: 2007.

3.2.3. PedidosYa.

La idea surge con tres chicos que decidieron crear una plataforma que reuniera la mayor cantidad de proveedores gastronómicos disponibles y se pudiera dividir por ubicación. Así, en 2009 nace PedidosYa como una plataforma que permite a los consumidores hacer pedidos por Internet.

Fundadores: Ariel Burschtin, Álvaro García y Ruben Sosenke

Lugar: sede principal Montevideo, Uruguay

Fecha: 2009.

3.2.4. ADN domicilios.

La idea surge con el inicio de la emergencia sanitaria de la pandemia de Covid-19, al inicio de la cuarentena obligatoria indicada por el gobierno nacional se decide crear la aplicación para ayudar a muchos negocios pequeños a que sus productos lleguen a más personas, funciona para las localidades en las que las aplicaciones de más renombre no cubren (Kennedy y Bosa).

Fundadores: Felipe Sanchez

Lugar: sede principal Bogotá, Colombia

Fecha: 2020

3.2.5. MacDonald's app.

Debido a la revolución de aplicaciones de entrega de comida, la gran cadena de comidas rápidas estadounidense decide lanzar dos aplicaciones inicialmente en E.E.U.U, la primera para el reparto de alimentos y la segunda para búsqueda de cupones de descuento esta última exclusivamente para Estados Unidos.

Al ser exclusiva del restaurante tuvo una gran acogida y esto conllevó a que la idea se llevara a más lugares del mundo en donde la cadena tiene más sucursales, se encuentra disponible para IOS y Android.

Fundadores: Richard y Maurice (Dick y Mac) McDonald y Ray A. Kroc.

Lugar: United States - San Bernardino (California).

Fecha: 1940.

3.2.6. El Corral app.

Siguiendo la línea de muchos restaurantes que tuvieron inconvenientes por el cierre parcial o total de sus instalaciones físicas por la pandemia de Covid-19 en el año 2020, el restaurante de comidas rápidas colombiano EL CORRAL decide lanzar la aplicación que lleva el mismo nombre al público.

El concepto es bastante conocido, se puede hacer un pedido de cualquiera de sus productos y son llevados hasta el lugar donde se solicitó por un mensajero propio de la empresa, después de la reapertura de los restaurantes se agregó la opción de hacer el pedido e ir a recogerlo por un mismo a la franquicia más cercana.

Fundadores: Pablo Emilio Bueno y Guillermo Calderón.

Lugar: Bogotá – Colombia.

Fecha: 8 de febrero de 1983

4. Especificación de requisitos de software (IEEE 830)

4.1. Perspectiva del producto

Se proyecta la generación de un sistema de generación de pedidos en la cafetería de la Universitaria Agustiniense para la reducción del tiempo en que una orden es recibida por el usuario final de la cafetería, para que así los mismos puedan invertir mejor los minutos libres entre o antes de clases.

4.2. Funcionalidad del producto

Los procesos que se llevarán a cabo para que el producto final muestre la funcionalidad requerida en las dos partes involucradas son:

4.2.1. Funciones aplicación de usuario.

Gestión de registro: en esta función se procederá a la realización de creación de usuario y posteriormente a guardar esta información dentro del gestor de base de datos ya mencionado anteriormente.

Esto se realizará mediante los procesos de: agregar datos, seleccionar una contraseña y posterior a ello ingresar nuevamente las credenciales.

Gestión de realización de pedidos: mediante la cual con esta función el usuario procederá a realizar el pedido a la cafetería de la universitaria agustiniana, con los pasos de ingreso de las credenciales, seleccionar el artículo o artículos que desea ordenar, dar creación de pedido y esperar a la confirmación por parte de la cafetería.

Gestión de visualización de últimos pedidos: en esta función el usuario podrá ingresar a la pestaña con este nombre a visualizar sus últimos pedidos, en esta opción podrá tomar como opción volver a pedir estos artículos o ver cuanto fue el precio.

Gestión visualización de productos de la cafetería: en esta función teniendo en cuenta el proceso de pedidos el usuario podrá ver: artículos que ofrece la cafetería tanto productos hechos como golosinas empacadas, los mismos están divididos en la página de pedidos.

4.2.2. Funciones aplicación de la cafetería.

Gestión de visualización de pedidos realizados en tiempo real: en esta función la cafetería de la Universitaria podrá visualizar los pedidos que los usuarios de la aplicación realizando, en la misma podrá aceptarlos o rechazarlos dependiendo de disponibilidad y horarios.

Gestión de precios: mediante esta función la cafetería tendrá completo control del manejo de los precios de los artículos disponibles, se realizará recomendación de modificación de los mismo antes de iniciar proceso de pedidos.

Gestión de Productos: esta función permitirá al restaurante hacer la modificación de los artículos dependiendo de lo que necesite, en la misma podrá: eliminar artículos, cambiarlos, crear o colocarlos no disponibles.

Gestión de arqueo de caja: esta función permitirá tener un control final de la caja de la cafetería, en donde podrá: controlar el ingreso y salida de efectivo, tener un registro del inventario.

4.3. Características de los usuarios

En la aplicación se intervendrán en total 2 usuarios que manipularán el producto final: usuario - cliente y usuario – cafetería. Cada uno tiene un perfil para la interacción final de la aplicación y que funcione correctamente:

4.3.1. Usuario – cliente.

Este perfil manejará la aplicación disponible para todos los miembros de la comunidad de la Universitaria Agustiniiana, podrá hacer uso de ella dentro de las instalaciones de la Universitaria y le permitirá realizar los pedidos, recoger los alimentos y consumir los artículos solicitados de la cafetería.

4.3.2. Usuario – cafetería.

Este perfil manejará el proceso de recibir, preparar, entregar, cobrar los artículos solicitados por los usuarios del primer perfil mediante una aplicación web independiente a la que solo ellos tendrán acceso, esta no está disponible en las tiendas de aplicaciones.

4.4. Restricciones

La aplicación se realizará en software de licencia abierta por dicho motivo no se deberá tener que hacer pagos por los programas elegidos para el desarrollo general tanto de la aplicación de usuario como la de la cafetería de la Universitaria Agustiniiana.

Se mantendrá un proceso de guardado de información automático por lo que no se tiene manejo manual en este caso.

Con el proceso de la cafetería de la Universitaria Agustiniiana cuando los usuarios de esta deban modificar artículos o precios lo realizarán por ellos mismos de manera manual en la aplicación, solo ellos tendrán inicialmente acceso a estas opciones.

4.5. Suposiciones y dependencias

La aplicación realiza el guardado de información en el gestor de bases de datos MySQL por lo que la dependencia inicial es el almacenaje de los datos personales de los usuarios. Los demás requisitos adjuntos podrán estar sujetos a modificaciones por lo que se está realizando la documentación de cambios en tiempo real.

Se debe realizar una capacitación inicial al personal de la cafetería para el correcto funcionamiento y se hará antes de lanzarla al público.

4.6. Requisitos específicos

Requisitos específicos usuario:

Tabla 6.

Requisitos E

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 1	Información de artículos	09.05.2020	Esencial
Descripción	Es necesario que la aplicación de usuario les muestre a los usuarios la información general de los artículos (precios, descripción y fotografía)		
Proceso	Esto se realizará de manera en que al abrir la aplicación de usuario e intentar hacer el pedido ya aparezcan los artículos.		

Nota. Autoría propia. Tabla de requisitos específicos I.

Tabla 7.

Requisitos E2

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 2	Registro de usuario	09.05.2020	Esencial
Descripción	Realización de un registro de lo contrario no se podrá acceder a los servicios.		
Proceso	El usuario de la cafetería y usuario de la aplicación deberán crear un usuario para ingresar a la aplicación.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos II.

Tabla 8.

Requisitos E3

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 3	Guardado de información	10.11.2020	Esencial

Descripción	Se debe realizar el guardado de la información de la creación de los usuarios y modificación de contraseñas.
Proceso	El usuario al cambiar el usuario y contraseña se debe realizar el cambio automático y guardar la información.

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos III.

Tabla 9.
Requisitos E4

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 4	Carrito de compras	10.11.2020	Esencial
Descripción	La aplicación deberá poder agregar la información al seleccionar el botón del carrito de compras.		
Proceso	El usuario al seleccionar el artículo dará clic en el carrito de compras y este agregará la información final.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos IV.

Tabla 10.
Requisitos E5

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 5	Resumen de compra	10.11.2020	Esencial
Descripción	El sistema de la aplicación deberá permitir hacer un resumen de la compra que se quiere realizar.		
Proceso	Al agregar todos los artículos al carrito de compras el sistema dará un breve resumen de la compra.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos V.

Tabla 11.
Requisitos E6

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad

RF 6	Crear orden	10.11.2020	Esencial
Descripción	La aplicación deberá permitir crear la orden después de agregados los artículos		
Proceso	Al agregar todos los artículos al carrito de compras el sistema permitirá crear la orden.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos VI.

Tabla 12.
Requisitos E7

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 7	Confirmación de la orden	10.11.2020	Esencial
Descripción	El sistema de la aplicación deberá realizar la confirmación de la orden luego de ser realizada.		
Proceso	Luego de ser creada la orden aparecerá la confirmación de la misma.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos VII.

Tabla 13.
Requisitos E8

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 8	Ordenes anteriores	10.11.2020	Esencial
Descripción	Luego de finalizada la orden se deberá mover a ordenes anteriores.		
Proceso	Luego de finalizada la orden se debe mover a ordenes anteriores para que quede el registro del pedido.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos VII.

Requisitos Específicos Cafetería:

Tabla 14.
Requisitos EC1

Requisitos Específicos

Código	Nombre	Fecha	Grado de necesidad
RF 1	Guardar información Ad	7.11.2020	Esencial
Descripción	Se deberá guardar la información de administrador		
Proceso	El administrador de la cafetería deberá crear el usuario y guardarlo.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos cafetería I.

Tabla 15.
Requisitos EC2

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 2	Artículos existentes	7.11.2020	Esencial
Descripción	La aplicación web deberá mostrar la información de los artículos de la cafetería.		
Proceso	Al ingresar en la aplicación de la cafetería deberá mostrar el listado de los artículos.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos cafetería II.

Tabla 16.
Requisitos EC3

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 3	Artículos modificados	7.11.2020	Esencial
Descripción	La aplicación web deberá mostrar los artículos modificados.		
Proceso	Al ingresar en la aplicación de la cafetería deberá mostrar los resultados de los artículos modificados		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos cafetería III.

Tabla 17.
Requisitos EC4

Requisitos Específicos			
------------------------	--	--	--

Código	Nombre	Fecha	Grado de necesidad
RF 4	Aceptar orden	7.11.2020	Esencial
Descripción	La aplicación web deberá mostrar la orden enviada y aceptarla.		
Proceso	Cuando el usuario envíe la orden deberá ser posible aceptarla desde la aplicación web.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos cafetería IV.

Tabla 18.
Requisitos EC5

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 5	Resumen de la orden	7.11.2020	Esencial
Descripción	La aplicación web deberá mostrar un resumen de la orden del usuario.		
Proceso	Se mostrará un resumen de la orden realizada por el usuario.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos cafetería V.

Tabla 19.
Requisitos EC6

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad
RF 6	Finalizar orden	7.11.2020	Esencial
Descripción	La aplicación web deberá permitir la finalización de la orden.		
Proceso	Cuando la orden sea entregada se deberá permitir la finalización del pedido.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos cafetería VI.

Tabla 20.
Requisitos EC7

Requisitos Específicos			
Código	Nombre	Fecha	Grado de necesidad

RF 7	Modificación de precios	7.11.2020	Esencial
Descripción	La aplicación web deberá permitir cambiar o modificar los precios.		
Proceso	Cuando sea necesario el administrador de la cafetería podrá modificar los artículos.		

Nota. Autoría propia. Tabla de especificaciones de los requisitos específicos cafetería VII.

4.6.1. Actores/roles.

Tabla 21.

Usuario Cliente

Actor	Usuario - Cliente
Descripción	El usuario deberá inicialmente inscribirse a la aplicación y así crear sus credenciales para que se permita el ingreso a la misma.
Características	El usuario será identificado por sus credenciales y de esta manera podrá acceder a los servicios de la aplicación.
Relación	Las credenciales del usuario serán validadas para el retiro de los alimentos en la cafetería
Referencias	<ol style="list-style-type: none"> 1. Inscripción 2. Ingreso de credenciales. 3. Navegación en la aplicación. 4. Realización de pedidos. 5. Retiro de pedidos.

Nota. Autoría propia. Tabla de roles de usuario-cliente.

Tabla 22.

Usuario Cafetería

Actor	Usuario - Cafetería
Descripción	El usuario deberá inicialmente inscribirse a la aplicación y así crear sus credenciales para que se permita el ingreso a la misma.
Características	El usuario será identificado por sus credenciales y de esta manera podrá acceder a los servicios de la aplicación.
Relación	Las credenciales del usuario serán validadas por los miembros de la cafetería
Referencias	<ol style="list-style-type: none"> 1. Inscripción 2. Ingreso de credenciales. 3. Navegación por el programa 4. Edición de los artículos 5. Manejo de los pedidos.

Nota. Autoría propia. Tabla de roles de usuario-cafetería.

4.6.2. Requisitos funcionales.

Tabla 23.

Req. Funcionales

Requisitos Funcionales			
Código	Nombre	Fecha	Grado de necesidad
RF 1	Ingreso aplicación		Esencial
Descripción	El usuario debe crear sus credenciales para permitirse su ingreso a la aplicación		
Proceso	Se usuario debe ingresar el usuario, contraseña y confirmar la misma, la aplicación al aceptar las credenciales lo redirigirá a la pantalla de inicio para que las vuelva a agregar		

Nota. Autoría propia. Tabla de requisitos funcionales 1.

Tabla 24.

Req. Funcionales 2

Requisitos Funcionales			
Código	Nombre	Fecha	Grado de necesidad
RF 2	Actualización de contraseña		Esencial
Descripción	Al crear las credenciales se da la opción al usuario de que pueda editar la contraseña o cambiarla si así lo desea.		
Proceso	El usuario indicará si se le olvido la contraseña o actualizar para que el proceso pueda ser culminado.		

Nota. Autoría propia. Tabla de requisitos funcionales 2.

Tabla 25.

Req. Funcionales 3

Requisitos Funcionales			
Código	Nombre	Fecha	Grado de necesidad
RF 3	Credenciales de ingreso		Esencial
Descripción	Siempre que el usuario desee ingresar a la aplicación para hacer un pedido debe ingresar sus credenciales		
Proceso			

	El usuario colocará en el campo usuario su código de la universidad y en el de contraseña colocará la que ya había seleccionado anteriormente
--	---

Nota. Autoría propia. Tabla de requisitos funcionales 3.

Tabla 26.

Req. Funcionales 4

Requisitos Funcionales			
Código	Nombre	Fecha	Grado de necesidad
RF 4	BD de la información general		Esencial
Descripción	Todos los cambios que se realizan en contraseña y usuario quedan almacenados dentro del gestor de bases de datos		
Proceso	El usuario no ve este proceso, solo se encarga de realizar la inscripción y cambios.		

Nota. Autoría propia Tabla de requisitos funcionales 4.

Tabla 27.

Req. Funcionales 5

Requisitos Funcionales			
Código	Nombre	Fecha	Grado de necesidad
RF 5	Arqueo de Caja		Esencial
Descripción	El proceso de arqueo se realizará en base a la información de ingreso de credenciales del administrador.		
Proceso	Este proceso se realizará con el usuario de administrador, creará el proceso para ver el informe del día.		

Nota. Autoría propia. Tabla de requisitos funcionales 5.

4.6.3. Diagrama de casos de uso.

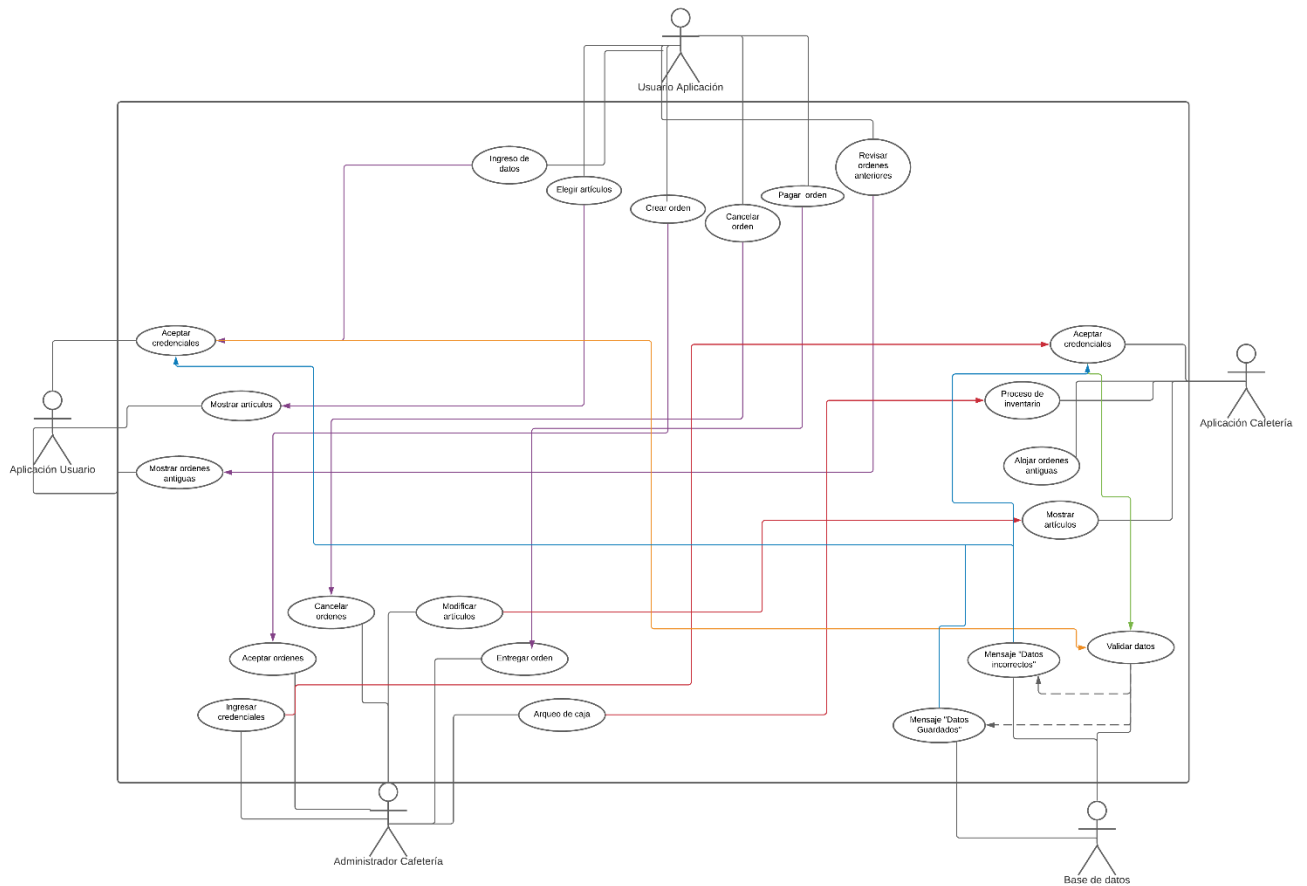


Figura 12. Caso de uso general. Autoría propia.

Código de color:

Morado: procesos salientes de actor (Usuario Aplicación)

Fucsia: procesos salientes de actor (Administrador cafetería)

Naranja: procesos salientes de actor (Aplicación usuario)

Verde: procesos salientes de actor (Aplicación cafetería)

Azul: procesos salientes de actor (Base de datos)

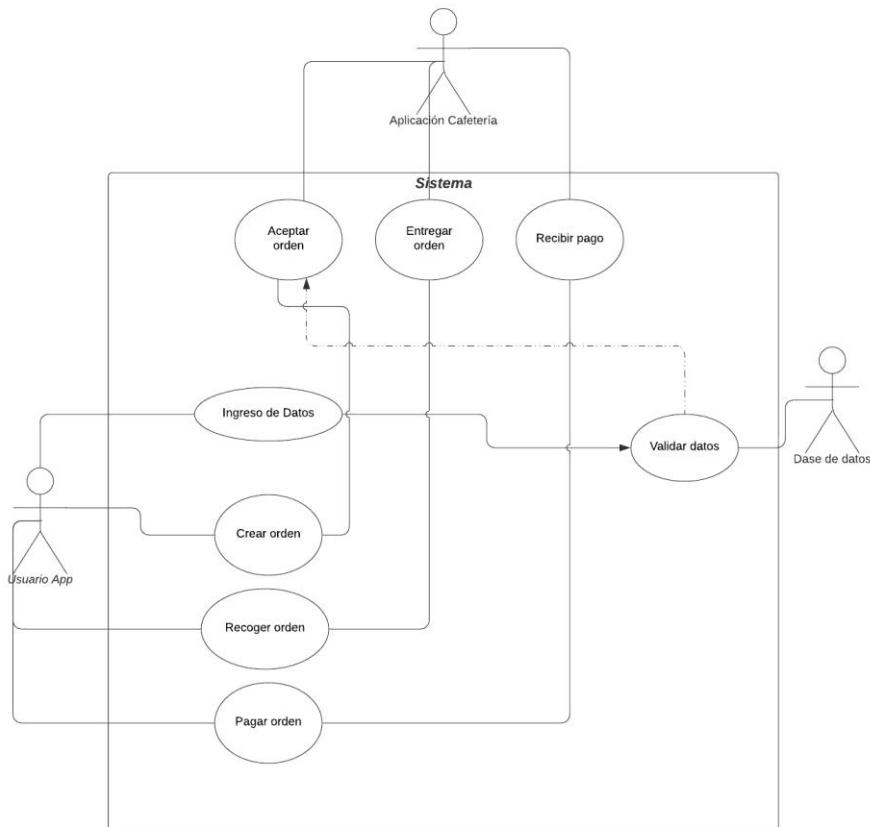


Figura 13. Caso de uso Usuario App y Cafetería. Autoría propia.

4.6.4. Especificaciones de los casos de uso.

Tabla 28.

Casos de uso 1

Identificación de Caso de Uso	
1.1 ID Caso	CD-230
1.2 Nombre	Caso de uso General
2. Histórico de caso de uso	
2.1 Autor	Diana García
2.2 Fecha de Creación	01 de octubre de 2020
2.3 Última Actualización	01 de octubre de 2020
2.4 Actualizado por	Diana García
3. Definición de Caso de Uso	
3.1 Descripción	
Mediante ese caso de uso se explica el proceso general de uso de la aplicación de usuario y la aplicación de la cafetería, creación de órdenes y fases generales	

3.2 Actores		
1. Usuario aplicación 2. Usuario cafetería 3. Base de datos 4. Aplicación Usuario 5. Aplicación cafetería.		
3.3 Precondiciones		
3.3.1 Usuario Autenticado: Para poder realizar el proceso de ingreso a cualquiera de las aplicaciones ya sea usuario o de cafetería se deben ingresar las credenciales que acrediten a la persona como miembro.		
3.4 Flujo Normal		
Contando con las precauciones el flujo normal será el siguiente:		
Paso	Actor	Sistema
1	El usuario ingresará las credenciales	La información se recibe
2	El usuario podrá ingresar a la aplicación para navegar en el sistema	El sistema mostrará la información
3	El usuario seleccionará los artículos que desee para crear la orden	El sistema guardará los artículos para la creación de orden
4	El sistema creará la orden	La base de datos guarda el pedido
5	El usuario podrá cancelar la orden si así lo desea, debe ser antes de que sea aceptada	El sistema cancelará la orden y la borrará del historial
6	El usuario esperará a que sea confirmada su orden	El sistema enviará confirmación una vez la cafetería indique la finalización de la orden
7	El usuario de la cafetería ingresará las credenciales de uso	La información se recibe
8	Esperará a que llegue un nuevo pedido o solicitud	El sistema enviará notificación cuando llegue un nuevo pedido
9	Aceptará o declinará el pedido	Se enviará confirmación
10	Preparará el pedido	El sistema queda en pausa

11	Se verá cuando se pueda recoger el pedido	El sistema se actualizará para que el usuario se dirija a recoger el pedido
12	El usuario se dirige a recoger el pedido	El sistema finaliza la orden.

Nota. Autoría propia. Tabla de casos de uso 1.

Tabla 29.

Casos de uso 2

Identificación de Caso de Uso		
1.1 ID Caso	CD-230	
1.2 Nombre	Caso de uso específico	
2. Histórico de caso de uso		
2.1 Autor	Diana García	
2.2 Fecha de Creación	01 de octubre de 2020	
2.3 Última Actualización	01 de octubre de 2020	
2.4 Actualizado por	Diana García	
3. Definición de Caso de Uso		
3.1 Descripción		
Mediante ese caso de uso se explica el proceso específico de creación de orden entre el usuario de la cafetería y el usuario de la aplicación.		
3.2 Actores		
1. Usuario aplicación 2. Usuario cafetería 3. Base de datos		
3.3 Precondiciones		
3.3.1 Usuario Autenticado: Para poder realizar el proceso de ingreso a cualquiera de las aplicaciones ya sea usuario o de cafetería se deben ingresar las credenciales que acrediten a la persona como miembro.		
3.4 Flujo Normal		
Contando con las precauciones el flujo normal será el siguiente:		
Paso	Actor	Sistema
1	El usuario ingresará las credenciales	La información se recibe

2	El usuario podrá ingresar a la aplicación para navegar en el sistema	El sistema mostrará la información
3	El usuario seleccionará los artículos que desee para crear la orden	El sistema guardará los artículos para la creación de orden
4	El sistema creará la orden	La base de datos guarda el pedido
5	El usuario esperará a que sea confirmada su orden	El sistema enviará confirmación una vez la cafetería indique la finalización de la orden
6	El usuario de la cafetería ingresará las credenciales de uso	La información se recibe
7	Esperará a que llegue un nuevo pedido o solicitud	El sistema enviará notificación cuando llegue un nuevo pedido
8	Aceptará o declinará el pedido	Se enviará confirmación
9	Preparará el pedido	El sistema queda en pausa
10	El usuario se dirige a recoger el pedido	El sistema finaliza la orden.

Nota. Autoría propia. Tabla de casos de uso 2.

4.7. Requisitos de rendimiento

Se contará con un servidor de base de datos en línea, donde los usuarios interactuarán con la aplicación de usuario y el programa de la cafetería, el mismo estará habilitado para recibir varias órdenes al mismo tiempo y que no se creen bugs. La base de datos contar con un respaldo de datos, por medio de back up diarios.

4.8. Restricciones de diseño

No se han realizado restricciones de diseño, se han mantenido dentro de los estándares de imagen institucional de la Universitaria Agustiniiana. Se debe tener en cuenta que la aplicación solo podrá ser abierta en dispositivos móviles y el programa de la cafetería solo será abierto por ellos en un computador.

4.9. Atributos del software del sistema

Dentro de la aplicación tanto la de usuario – Cliente y usuario – Cafetería se consideran como atributos:

Fiabilidad: la aplicación cumple con las determinaciones que se plantean dentro de este documento por lo que se dará muestra de ello en las pruebas finales.

Seguridad: los datos compartidos en la aplicación no serán compartidos, divulgados, usados ni editados.

Disponibilidad: la aplicación de usuario – Cliente estará disponible para los sistemas operativos más usados por los miembros de la comunidad de la Universitaria Agustiniiana.

Rendimiento: se realizarán las pruebas completas para garantizar el rendimiento para los usuarios como la cafetería.

Confiabilidad: este proyecto está realizado de estudiantes para la comunidad así que se asegura un proceso de confiabilidad.

5. Diseño del software (ISO -12207-1)

5.1. Diseño de la arquitectura de software

Para el diseño de la arquitectura se decidió usar el modelo C4, esto debido a que por sus niveles jerárquicos en cuanto a los diagramas indican, podemos brindar una vista específica de lo que se espera hacer con el proyecto y así llegar a varias audiencias.

El modelo de diseño de arquitectura c4 es usado para diagramación de código de manera sencilla y esto lo podemos visualizar en su página oficial como:

El modelo C4 se creó como una forma de ayudar a los equipos de desarrollo de software a describir y comunicar la arquitectura de software, tanto durante las sesiones de diseño iniciales como al documentar retrospectivamente una base de código existente. Es una forma de crear mapas de su código, en varios niveles de detalle, de la misma manera que usaría algo como Google Maps para acercar y alejar un área que le interesa.

5.1.1. Diagrama de contexto del sistema.

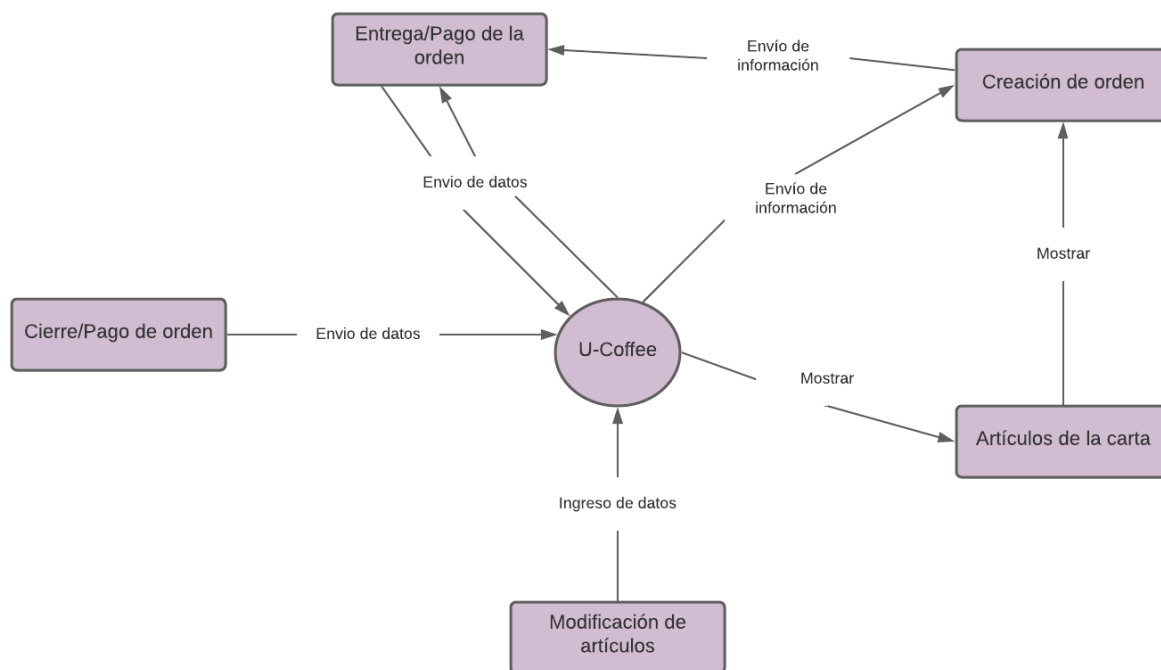


Figura 14. Diagrama de contexto de sistema. Autoría propia.

5.1.2. Diagrama de contenedor.

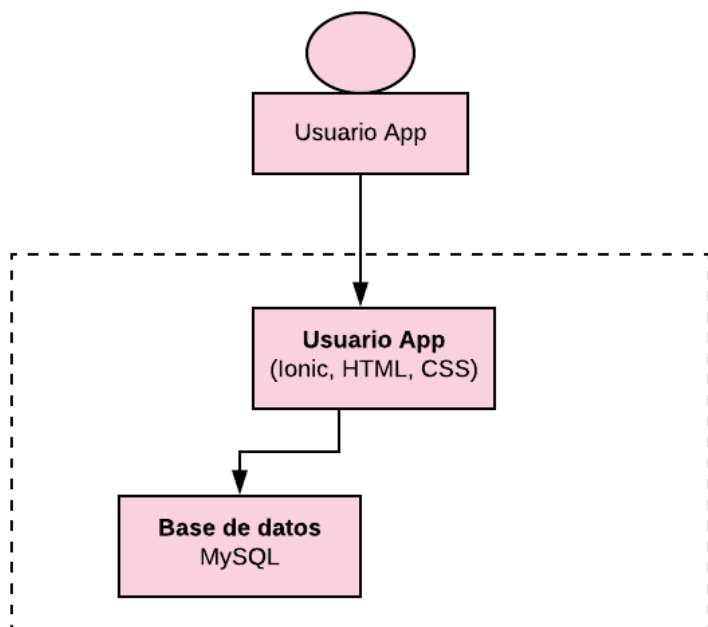


Figura 15. Diagrama de contenedor de usuario. Autoría propia.

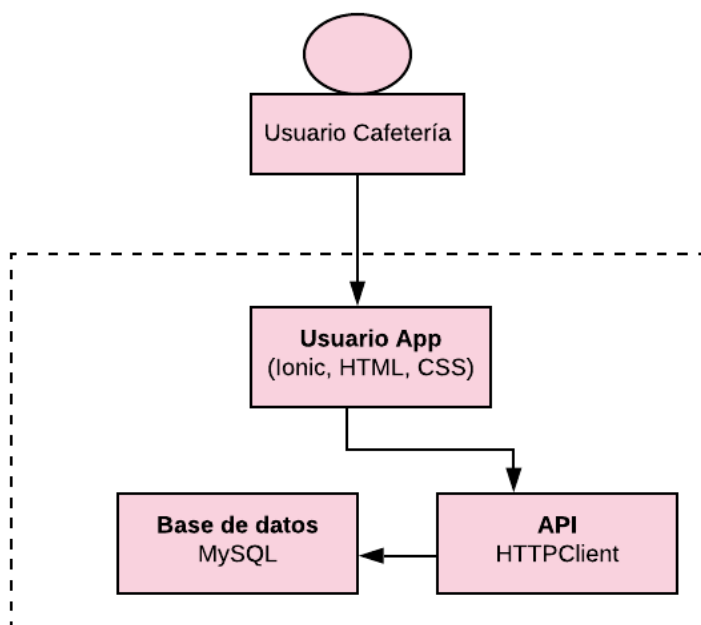


Figura 16. Diagrama de contenedor de cafetería. Autoría propia.

5.1.3. Diagrama de componentes.

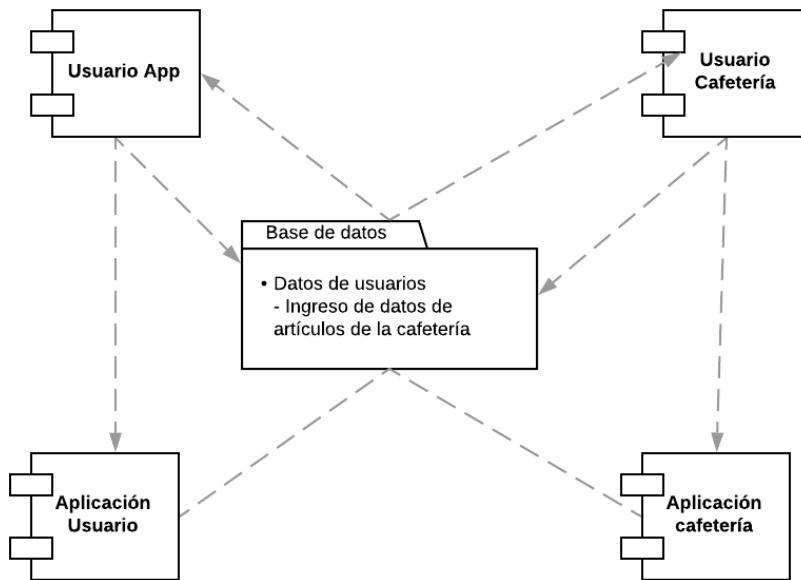


Figura 17. Diagrama de componentes. Autoría propia.

5.2. Diseño detallado del software

5.2.1. Diagrama de clases.

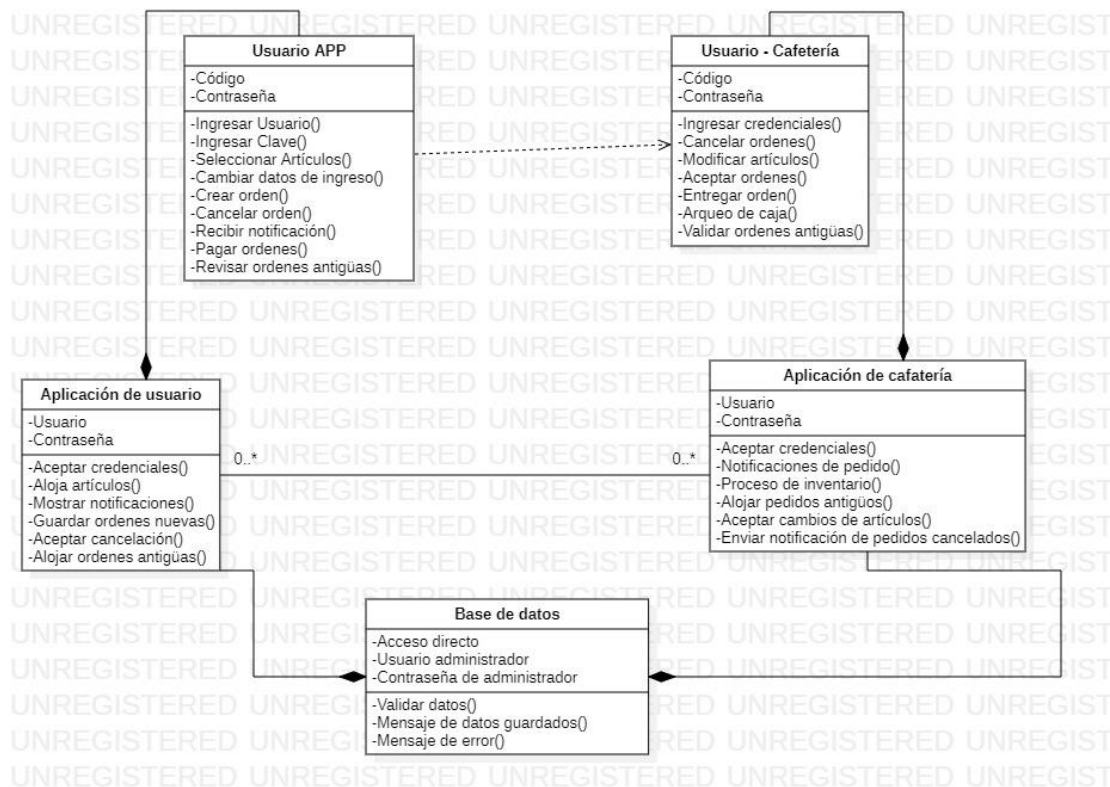


Figura 18. Diagrama de clases. Autoría propia.

5.2.2. Diagrama de paquetes.

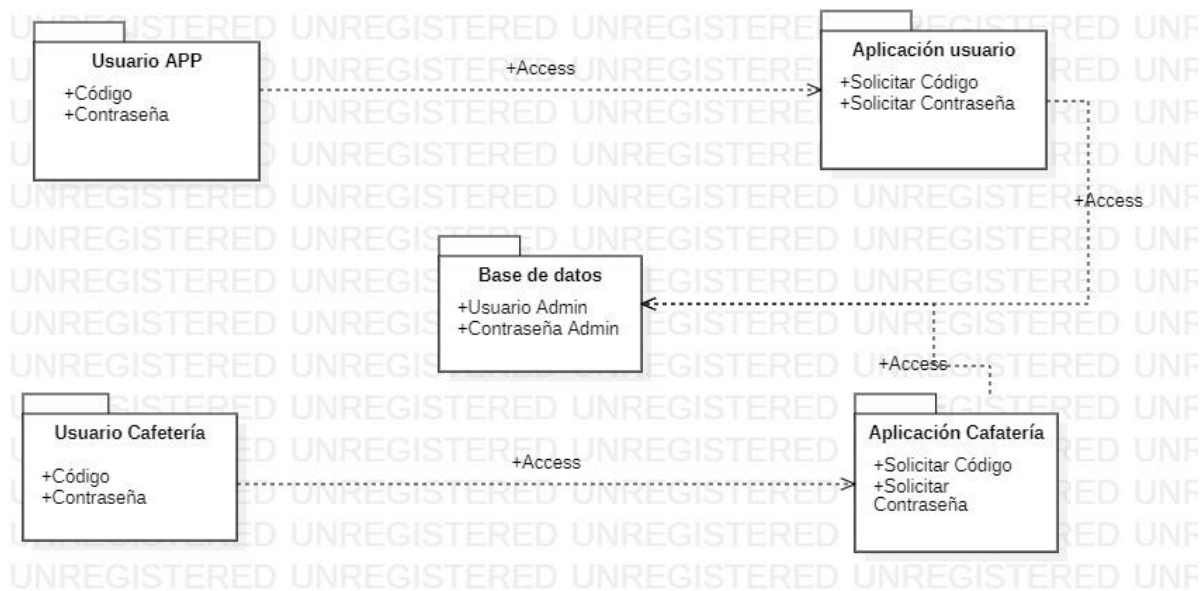


Figura 19. Diagrama de paquetes. Autoría propia.

5.2.3. Diagramas de despliegue.

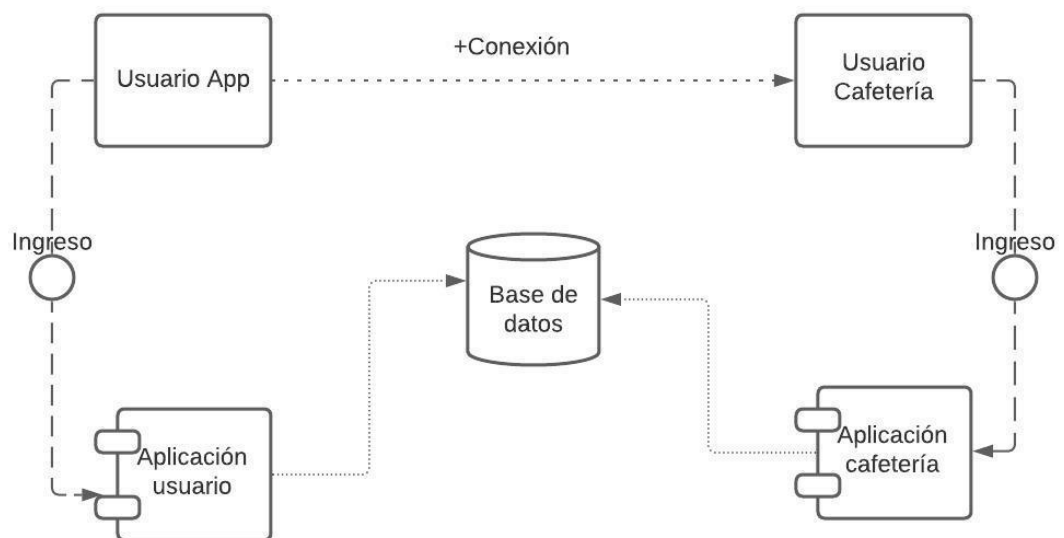


Figura 20. Diagrama de despliegue. Autoría propia.

5.3. Diseño de la Interfaz

5.3.1. Interfaz gráfica de usuario.

The image shows two side-by-side mobile app screens. The left screen, titled 'Ingreso', features the 'Coffee' logo at the top, followed by input fields for 'Número de Documento' and 'Contraseña'. Below these are two orange buttons: 'INGRESAR' and '¡REGÍSTRATE!'. The right screen, titled 'Registro', has a back arrow and a title bar. It contains input fields for 'Número de documento', 'Nombre(s)', 'Apellido(s)', 'Correo', 'Contraseña', and 'Confirmar contraseña'. An orange 'ENVIAR' button is at the bottom. A shared bottom navigation bar includes icons for 'Mis Pedidos', 'Menú', and 'Recoge el pedido'.

Figura 21. Interfaz gráfica de usuario. Autoría propia.


The image displays two screenshots of a web-based management interface. The top screenshot shows a 'Pedidos sin alistar' (Orders to be prepared) table. The bottom screenshot shows a 'Datos administrador' (Admin Data) modal for deleting a product.

Pedidos sin alistar

Cliente	Descripción	Total	Alistar pedido
Daniel Torres	Cafe Expansado	\$ 3.000	<input type="radio"/>
Daniel Torres	Petito caliente.Pizza	\$ 6.000	<input type="radio"/>
Daniel Torres	Pizza	\$ 4.500	<input type="radio"/>
Daniel Torres	Pizza	\$ 4.500	<input type="radio"/>
Daniel Torres	Pizza	\$ 4.500	<input type="radio"/>
Daniel Torres	Sandwich.Pizza	\$ 6.500	<input type="radio"/>

Datos administrador

Ingrese los datos del administrador para eliminar este producto:

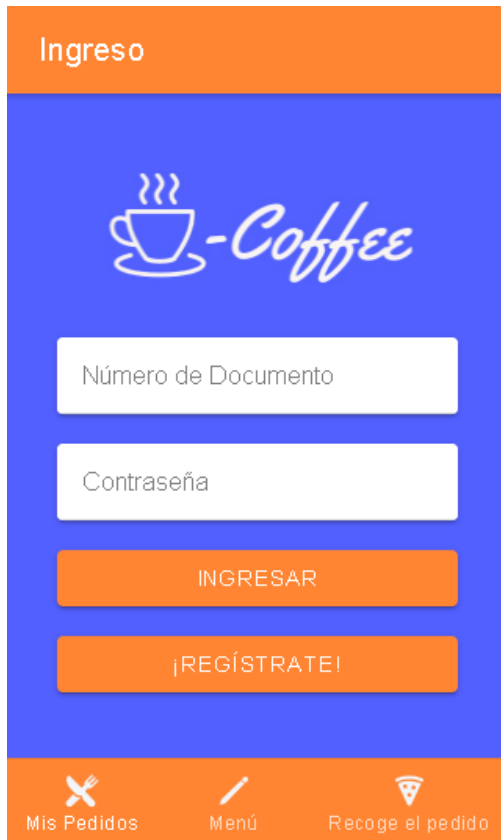
 Almojábana
Valor: \$ 2.900

Usuario: _____
Contraseña: _____

Figura 22. Interfaz gráfica de cafetería. Autoría propia.

5.3.2. Interfaces de entrada.

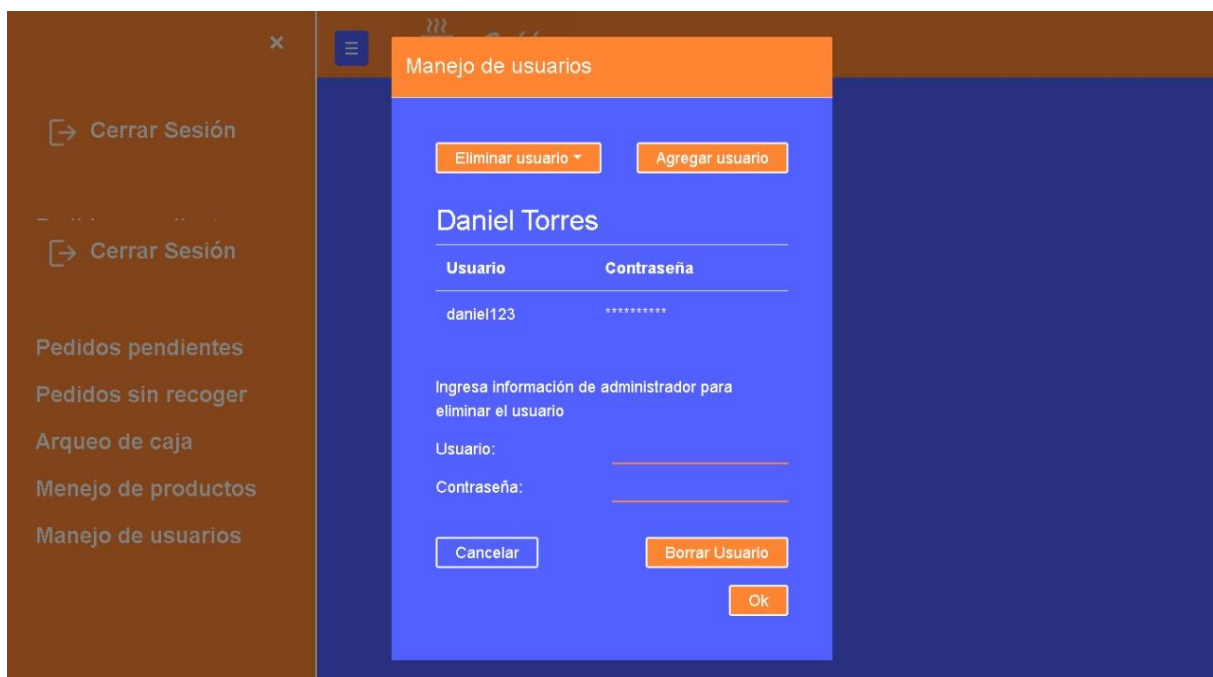
Aplicación de usuario:



The screenshot shows a mobile application interface for user login. At the top, there is an orange header with the word "Ingreso". Below this is a blue background featuring a white coffee cup icon with steam and the text "-Coffee" in a cursive font. There are two white input fields: "Número de Documento" and "Contraseña". Below the input fields are two orange buttons: "INGRESAR" and "¡REGÍSTRATE!". At the bottom, there is an orange navigation bar with three icons and labels: a fork and knife for "Mis Pedidos", a pencil for "Menú", and a Wi-Fi symbol for "Recoge el pedido".

Figura 23. Interfaz de entrada de usuario. Autoría propia.

Aplicación cafetería:



The screenshot shows a desktop application interface for user management. On the left is a dark brown sidebar with a menu containing: "Cerrar Sesión", "Cerrar Sesión", "Pedidos pendientes", "Pedidos sin recoger", "Arqueo de caja", "Menejo de productos", and "Manejo de usuarios". The main content area has a blue background and is titled "Manejo de usuarios" in an orange header. It contains two orange buttons: "Eliminar usuario" and "Agregar usuario". Below these is the name "Daniel Torres". A table shows user details:

Usuario	Contraseña
daniel123	*****

Below the table, there is a prompt: "Ingresa información de administrador para eliminar el usuario". This is followed by two input fields labeled "Usuario:" and "Contraseña:". At the bottom, there are three buttons: "Cancelar", "Borrar Usuario", and "Ok".

Figura 23. Interfaz de entrada de cafetería. Autoría propia.

5.3.3. Interfaces de salida.

Aplicación de usuario:



Figura 24. Interfaz de salida de usuario. Autoría propia.

Aplicación cafetería:



Figura 25. Interfaz de salida de la cafetería. Autoría propia.

6. Implementación

6.1. Plataformas de desarrollo

Tanto la aplicación usuario y cafetería fueron desarrolladas en Ionic Framework, Angular, HTML5, CSS, para la base de datos se usó MySQL Workbench, como editor de código se maneja visual Studio Code y para creación del paquete de instalación de la aplicación.

6.2. Base de datos



Figura 26. Base de datos. Autoría propia.

6.3. Infraestructura de hardware y redes

6.3.1. Aplicación cafetería.

La aplicación que administra los pedidos que realizan los usuarios será una aplicación Web, esta estará alojada en un servidor. Para acceder a sus funciones se solicitará un usuario y contraseña que será suministrado por la persona encargada y autorizada para crear, actualizar y eliminar dichos usuarios.

La página web ‘<https://u-coffee.000webhostapp.com/>’ es donde estará alojada la aplicación dedicada a administrar toda la información de U-Coffee.

6.3.2. Aplicación usuario.

Para la descarga de la aplicación móvil se deben seguir los siguientes pasos:

Ingresar al Dominio ‘<https://u-coffee.000webhostapp.com/>’.

En la parte inferior aparecerá el ícono de Android y IOS.

Debe hacer click en el ícono de la plataforma en la cual será instalada la aplicación para iniciar la descarga.

Una vez finalizada la descarga debe dirigirse al directorio donde el archivo APK fue descargado para iniciar la instalación.

7. Pruebas del software

7.1. Pruebas del software

En donde las mismas fueron enviadas con un programador para que el mismo nos diera el aval de los requisitos funcionales de los programas.

Tabla 30.
Pruebas 1

Pruebas de la aplicación de Usuario			
No.	Requisito Funcional	Cumple	No Cumple
RQF1	El sistema deberá guardar la información de usuario	x	
RQF2	El sistema deberá mostrar la información de los artículos	x	
RQF3	El sistema deberá permitir guardar la información al presionar el "Carrito de Compras"	x	
RQF4	El sistema deberá permitir crear la orden	x	
RQF5	El sistema deberá mostrar un resumen de la compra	x	
RQF6	El sistema deberá mostrar la confirmación de la orden	x	
RQF7	El sistema deberá mover la orden cuando esté finalizada a ordenes anteriores.	x	

Nota. Autoría propia. Tabla de pruebas de usuario.

Tabla 31.
Pruebas 2

Pruebas de la aplicación de Cafetería			
No.	Requisito Funcional	Cumple	No Cumple
RQF1	El sistema deberá guardar la información del administrador	x	
RQF2	El sistema deberá mostrar la información de los artículos existentes	x	
RQF3	El sistema deberá mostrar la información de los artículos modificados	x	
RQF4	El sistema deberá permitir aceptar la orden	x	
RQF5	El sistema deberá mostrar un resumen de la orden	x	
RQF6	El sistema deberá mostrar la confirmación de la orden	x	
RQF7	El sistema deberá permitir dar por finalizada la orden	x	
RQF8	El sistema deberá permitir modificar los precios de los artículos	x	

Nota. Autoría propia. Tabla de pruebas de cafetería.

7.2. Pruebas de usabilidad

Se decidió tomar como base para pruebas de usabilidad la evaluación Heurística y tomaremos como definición la brindada por Gonzáles, Pascual y Lorés (1997):

La palabra heurística procede etimológicamente de la palabra griega “euriskein” que procede de “eureka”, un vocablo que significa hallar o encontrar, Este vocablo fue exclamado por Arquímedes en un famoso episodio sin bases históricas. El diccionario de la Real Academia Española define a la palabra “heurística” como:

- Técnica de la indagación y del descubrimiento.
- Busca o investigación de documentos o fuentes históricas.
- En algunas ciencias, manera de buscar la solución de un problema mediante métodos no rigurosos, como por tanteo, reglas empíricas, etc.

Partiendo de la definición anterior, la Interacción Persona Ordenador (IPO) presenta da la Evaluación Heurística (EH) como un método de evaluación de la usabilidad por inspección que debe ser llevado a cabo por evaluadores expertos a partir de unos principios (denominados “heurísticos”) previamente establecidos. Por ser una técnica de evaluación de la usabilidad. La EH tiene como objetivo el medio la calidad de la interfaz de cualquier sistema interactivo en relación con su facilidad para ser aprendido y usado por un determinado grupo de usuarios en un determinado contexto de uso [ISO98, UNET06]. (p. 3).

Se tomarán como principales usuarios para la muestra a:

Expertos en usabilidad: se tomaron como principales expertos a los creadores de la aplicación, ya que al conocer cada parte del diseño e implementación de la aplicación y programa de la cafetería podían tener muy presente el hecho de los errores que se podrían presentar a lo largo de las pruebas.

Los desarrolladores: dentro de este proceso inicial se tomaron como principales desarrolladores a nuestros compañeros de sexto (6) semestre de la carrera (Tecnología en desarrollo de software), esto inicialmente tomado por el conocimiento que ellos han adquirido y que se tomaron este proceso de una manera muy profesional.

Los usuarios potenciales: dentro de los usuarios tomamos a varias personas entre compañeros, familiares y personas externas (no conocidos por ninguno personalmente), de esta manera se tienen varios datos para confirmar el uso de la aplicación.

Para el desarrollo de las pruebas se decidió crear 3 formatos de prueba, los cuales consistieron en:

Prueba de Desarrolladores: Dentro de esta prueba el grupo de desarrolladores elegidos tuvieron un proceso de evaluación de dos partes:

Diseño: En donde se evaluará la parte de diseño general de la aplicación.

Testing: En donde se evaluará el proceso de cargue, desarrollo y generalidad sobre la usabilidad de la aplicación de usuario.

Figura 27. Pruebas [Desarrolladores]. Autoría propia.

Prueba de Usuario: Dentro de esta prueba los usuarios de la aplicación, después de usarla, podrán evaluar en dos aspectos importantes la aplicación:

Diseño: Por medio de la cual, los usuarios podrán calificar el diseño general de la aplicación.

Uso general: Por medio de la cual, el usuario podrá evaluar el uso de la aplicación, con cargue, desarrollo y generalidades de esta.

Figura 28. Pruebas [Usuarios]. Autoría propia.

Prueba de la cafetería: Por medio de la cual, el administrador de la cafetería en donde se implementó la aplicación, en la cual se evaluarán dos aspectos:

Diseño: Por el medio de la cual, se evaluó el diseño general, distribución y otros aspectos.

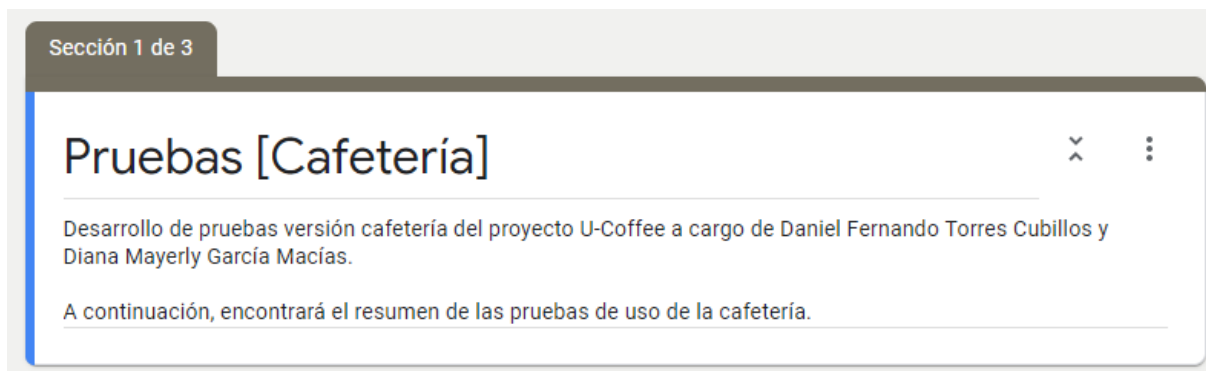


Figura 29. Pruebas [Cafetería]. Autoría propia.

De igual manera todas las pruebas tendrán una parte de sugerencias finales o “Feedback” en donde todos los miembros dependiendo de las características podrán brindar o no (no será obligatorio) una sugerencia final de la prueba.

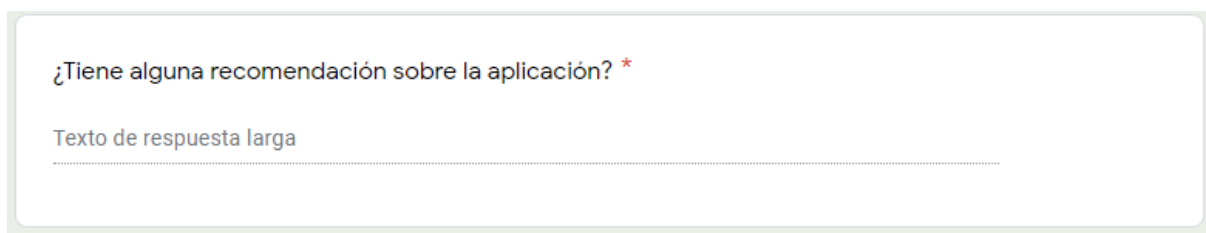
The image shows a feedback form with a light green border. The question is '¿Tiene alguna recomendación sobre la aplicación? *'. Below the question is a text input field with the placeholder text 'Texto de respuesta larga'.

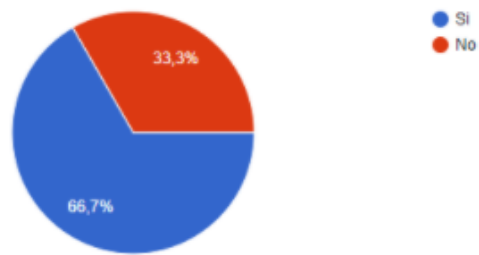
Figura 30. Sugerencias pruebas. Autoría propia.

7.2.1. Estadística de pruebas.

Pruebas de desarrolladores: Diseño

¿Considera que el diseño de la aplicación está acorde con la entidad corporativa de la Universitaria Uniagustiniana?

3 respuestas



Al revisar la aplicación ¿Considera ustedes que los colores combinan y son a gusto a la vista?

3 respuestas



Figura 31. Resultados pruebas desarrolladores [Diseño I]. Autoría propia.

¿Considera usted que el logo de la aplicación va acorde con la idea de la aplicación?

3 respuestas



¿Considera que la aplicación es amigable con el usuario?

3 respuestas

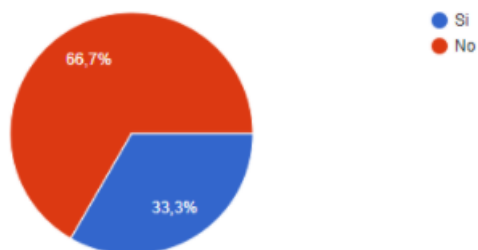


Figura 32. Resultados pruebas desarrolladores [Diseño II]. Autoría propia.

Pruebas de desarrolladores: Testing

¿Al ingresar a la aplicación la misma se puso lenta o se salió?

3 respuestas



¿Cuándo realizó el pedido le guardó los artículos que seleccionó?

3 respuestas



Figura 33. Resultados pruebas usuario [Testing I]. Autoría propia.

¿En el resumen del pedido le aparece los precios correctos y los artículos correctos?

3 respuestas



¿Al ser aceptado su pedido le apareció en la pestaña de "pedidos pendientes"?

3 respuestas



Figura 34. Resultados pruebas usuario [Testing II]. Autoría propia.

Pruebas de usuario: Diseño general

¿Al revisar la aplicación considera que de manera general el diseño es amigable?

12 respuestas



¿Considera usted que la aplicación representa a la Universitaria Uniagustiniana?

12 respuestas



¿Considera que la aplicación es sencilla de utilizar teniendo en cuenta la distribución realizada?

12 respuestas

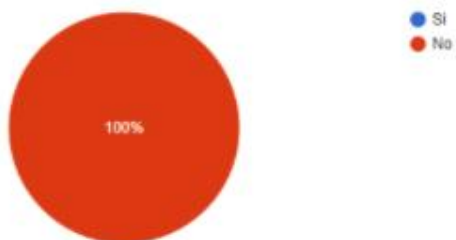


Figura 35. Resultados pruebas usuario [Diseño]. Autoría propia.

Pruebas de usuario: uso general

¿Al momento de crear un usuario en la aplicación tuvo algún inconveniente?

12 respuestas.



¿Tuvo problemas con la selección de los artículos en la aplicación?

12 respuestas.

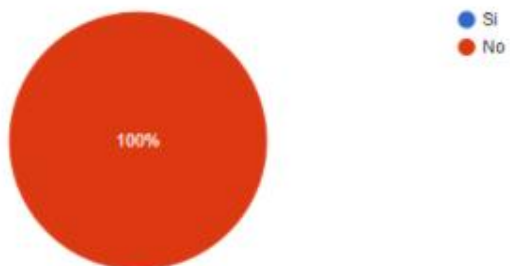


Figura 36. Resultados pruebas usuario [Uso general I]. Autoría propia.

¿Se agregaron todos los artículos que deseaba en el pedido?

12 respuestas.



¿Su pedido apareció como confirmado?

12 respuestas.



Figura 37. Resultados pruebas usuario [Uso general II]. Autoría propia.

¿El precio de los artículos fue el mismo que apareció en el pedido?

12 respuestas



¿La aplicación le pareció fácil de comprender y de usar?

12 respuestas



Figura 38. Resultados pruebas usuario [Uso general III]. Autoría propia.

Pruebas de cafetería: diseño

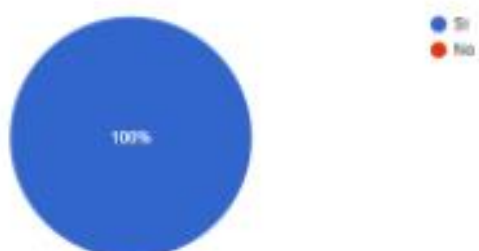
¿Cree usted que el diseño general de la interfaz es amigable?

1 respuesta



¿Considera que la aplicación es sencilla de utilizar teniendo en cuenta la distribución realizada?

1 respuesta



¿Considera que el proceso de edición de los artículos es sencillo?

1 respuesta



Figura 39. Resultados pruebas cafetería [Diseño]. Autoría propia (2020).

Prueba de cafetería: Uso general

¿Le arrojó algún error el programa cuando llegó una nueva orden?

1 respuesta



¿El programa se puso lento en más de una ocasión?

1 respuesta



¿Algún artículo no quedo guardado?

1 respuesta



Figura 4. Resultados pruebas cafetería [Uso general]. Autoría propia (2020).

Conclusiones

Dentro del desarrollo del proyecto como principal objetivo, nos habíamos planteado el aprendizaje y la relación de los conocimientos adquiridos durante los últimos tres años en la carrera de tecnología en desarrollo de software.

El hecho de culminar este proyecto y dar por finalizada esta etapa nos deja como enseñanza principal el proceso que llevamos a cabo durante casi un año, en donde hicimos sin quererlo una investigación que partió desde el día 0 (cero) cuando ingresamos al primer semestre y visualizamos el problema de la compra de artículos en la cafetería, en donde vivimos de primera mano las largas filas y la demora en la entrega, adicional a la falta de ciertos alimentos debido a que se agotaban por la alta demanda.

Tener la ayuda de nuestros compañeros de carrera y otros que si bien no eran de la facultad fueron de mucha ayuda para hacer un análisis específico, en donde salieron las preguntas respondidas en este documento: ¿Por qué se hará tanta fila en las horas “pico”? ¿Qué pasa si hacemos una aplicación de comidas para la Universitaria Agustiniana? ¿Qué pasa si la cafetería sabe cuántas personas van a pedir un mismo artículo? ¿Podrán dar abasto? ¿Funcionará?

Sobre todo, la última fue una que siempre tuvimos presente, lo pudimos visualizar en su implementación y en las pruebas realizadas anteriormente, este proceso culmina con todas nuestras preguntas resueltas y con los objetivos cumplidos.

Referencias

- C4. (s.f.). Maps of your Code. C4. Recuperado de: <http://c4model.com/>
- Carrasco, M. Ocampo, W. Ulloa, L. Azcona, J (2019). Metodología Híbrida de Desarrollo de Software Combinando Xp y Scrum. Revista Científica Multidisciplinaria Mikarimin, volumen (5), 112- 115.
- Correa, M. (2013). Composición del sistema operativo móvil IOS de Apple y el hardware y software que lo utilizan. (Trabajo de grado, Universidad Católica de Pereira). Recuperado de: <https://repositorio.ucp.edu.co/bitstream/10785/1984/1/CDMIST81.pdf>.
- Cruz Alonso, R. Gopar, L. López, I. Moya, E. (2008). Android. (Investigación, Universidad Tecnológica de la Mixteca). Recuperado de: <http://www.utm.mx/~caff/poo/AndroidIntro.pdf>
- Diccionario EcuRed. (2018a). App. EcuRed. Recuperado de: https://www.ecured.cu/App#Tipos_de_Aplicaciones_M.C3.B3viles
- Diccionario EcuRed. (2018b). App. EcuRed. Recuperado de: https://www.ecured.cu/Bases_de_datos
- Diccionario EcuRed. (2018c). App. EcuRed. Recuperado de: https://www.ecured.cu/P%C3%A1gina_Web
- González, M. Pascual, A. Lorés, J. (1997). Evaluación Heurística. Pontificia Universitat de Lleida, Lleida, España.
- Leterier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). cyta.com.ar. Recuperado de: www.cyta.com.ar/ta0502/v5n2a1.htm
- Muradas, Y. (2018). Metodologías y herramientas. [Grafico]. Recuperado de URL: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>
- PHP. (s.f.). ¿Qué es PHP?, PHP. Recuperado de: <https://www.php.net/manual/es/intro-what-is.php>
- Trigas, M. (s.f.). Gestión de Proyectos Informáticos. openaccess.uoc.edu. Recuperado de: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- Trigas, M. (s.f.). Gestión de Proyectos Informáticos. [Grafico]. Recuperado de URL: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>