

Aplicaciones inteligentes S.A de C.V.
“App para control de puesto de
antojitos mexicanos”.
Versión 1.0.

Ingenieros: Jose Alberto Cahuich
García.

Carlos Contreras Sáenz.

Angélica Noemí Guerrero Piña.

Ricardo Daniel Gasca López

Android_Studio.SA de CV

Fecha: 08/06/2020

Tabla de contenido

1. Hoja de control del documento	4
2. Historial de revisiones	4
3. Equipo de desarrollo	6
4. Vistas general de proyecto	7
Introducción.....	7
Reconocimiento del problemática	7
Justificación	7
Objetivos.....	7
Objetivos específicos.....	8
5. Análisis	8
Modelo.....	8
Modelo lineal secuencial.....	8
Metodología	11
Análisis de factibilidades	14
Diagrama de actividades	16
Diagrama de flujo de datos	16
Diagrama de clases	17
Diagrama de secuencia	17
Diccionarios de datos	18
6. Estructura lógicas de la información.....	18
Diseño	18
Diseño de la arquitectura del sistema	20
Prototipos de interfaces	22
7. Componentes y desarrollo SW	22
Programación.....	22
8. Gestión de riesgos.....	24
9. Verificación de la calidad	25
Pruebas	25
Normativas y protocolos.....	25
10. Entregables del proyecto	27

11. Catálogo de imágenes	27
---------------------------------------	----

12. Referencias	28
------------------------------	----

1.Hoja de control del documento

Título	Aplicaciones inteligentes S.A de C.V.
Descripción	Nombre genérico de los bocadillos que se comen en ambientes informales o antes de la comida y que se venden, por lo general, en puestos improvisados. Casi siempre están hechos de tortilla o masa de maíz. En los restaurantes suelen servirse como entremés. Un antojito puede ser también la comida principal, el desayuno o la cena, y no sólo un alimento entre comidas. Cada región tiene sus antojitos típicos y no puedes ubicar Avenida unión de colonos mz 10 lt 1, San Miguel Teotongo, C.P.09630, Alcaldia.Iztapalapa.
Autor	Jose Alberto Cahuich Garcia
Responsable	Carlos Contreras Sáenz.
Organización	Android_Studio.SA de _CV
Categoría	Desarrollo de software
Palabras	Punto de venta
Versión	1
Estadísticas	no
Edición	3

2.Historial de revisiones

Fecha	Descripción	Autor	Firma
27/02/2020	Nombres Nombre de empresa Llenar hoja de control Llenar historial de revisiones	Jose Alberto Cahuich García.	Carlos Contreras Sáenz.
12/03/2020	Llenar equipo de control Vista general de proyecto Modificar historial de revisiones Introducción	Jose Alberto Cahuich García.	Carlos Contreras Sáenz.
08/06/2020	Análisis. Modelo. Modelo lineal secuencial.	Jose Alberto Cahuich García.	Carlos Contreras Sáenz.

Departamento: Análisis.

	<p>Metodología.</p> <p>Análisis de factibilidades.</p> <p>Diagrama de actividades.</p> <p>Diagrama de flujo de datos.</p> <p>Diagrama de clases.</p> <p>Diagrama de secuencia.</p> <p>Diccionario de datos.</p> <p>Estructura lógicas de la información.</p> <p>Diseño.</p> <p>Diseño de la arquitectura del sistema.</p> <p>Prototipos de interfaces.</p> <p>Componentes y desarrollo SW.</p> <p>Programación.</p> <p>Gestión de riesgos.</p> <p>Verificación de la calidad.</p> <p>Pruebas.</p> <p>Normativas y protocolos.</p> <p>Entregables del proyecto.</p> <p>Catálogo de imágenes.</p> <p>Referencia.</p>		
--	--	--	--

3. Equipo de desarrollo

Nombre	Jose Alberto Cahuich Garcia.
Cargo	Análisis: Examen detallado de una cosa para conocer sus características o cualidades, o su estado, y extraer conclusiones, que se realiza separando o considerando por separado las partes que la constituyen.
Teléfono	5528308449
Correo	josecahuich.alu@itiztapalapa3.edu.mx

Nombre	Carlos Contreras Sáenz.
Cargo	Diseño: Conjunto de estudios necesarios para conseguir el título de diseñador y ejercer esa actividad.
Teléfono	5578866156
Correo	carloscontreras.alu@itiztapalapa3.edu.mx

Nombre	Angélica Noemí Guerrero Piña.
Cargo	Desarrollo: Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades.
Teléfono	5530719058
Correo	Angelicaguerrero.alu@itiztapalapa3.edu.mx

Nombre	Ricardo Daniel Gasca López.
Cargo	Implementación: Constituye la realización de determinados procesos y estructuras en un sistema. Representa así la capa más baja en el proceso de paso de una capa abstracta a una capa más concreta.
Teléfono	5617842676
Correo	ricardogasca.alu@itiztapalapa3.edu.mx

4. Vistas general de proyecto

Introducción

Los antojitos son un tipo de comida de la calle que se suele tomar como aperitivo y forma parte de la cultura mexicana. Por lo regular es un alimento a base de maíz, rico en grasas (muchas veces frito) y acompañado de una salsa de chile, y generalmente es parte de una comida rápida e informal.

Reconocimiento del problemática

A los propietarios del negocio “Antojitos Mexicanos” les interesa brindar una mejor atención a su clientela en cuanto a pagos de lo consumido de una forma rápida y sin fallas de cálculo. De parte interna llevar un buen registro de mercancía. Vendida, el margen de ganancia, el margen de perdida y una estadística para compra de su siguiente mercancía (próxima producción).

Justificación

La razón por la cual escogí este tema es porque para mí la gastronomía mexicana es una de los más importantes y uno al ser mexicano debes de valorar lo que se tiene. También otra cosa por lo que escogí este tema es que para toda aquella persona que vea el apps se le dará a entender que debe consumir comida mexicana y productos mexicanos para deshacernos de algo que para mí es una de las peores cosas del mundo como lo es la comida rápida o el comida rápida ya sea en su consumo y en su producción.

Objetivos

- Ofrecer la mejor atención a nuestros clientes de forma atenta y confiable.
- Darle importancia a cada uno de nuestros clientes pues de ellos depende nuestro éxito.
- Dar los mejores precios y más adecuados para nuestros platos.
- Dar información sobre la situación, facilidades y precios.

Objetivos específicos

- Desarrollar una aplicación móvil para poder atender mayor número de órdenes en menor tiempo y con mayor éxito.
- Tener una estadística impecable de clientes atendidos, productos más vendidos, ganancias y pérdidas.
- La información obtenida por el móvil debe ser almacenada para saber el éxito de Antojitos Mexicanos “El porvenir” y con esto influir a abrir más sucursales.

5. Análisis

Modelo

El modelo para desarrollo de software (aplicación móvil) es una representación abstracta de un proceso. Cada modelo representa un proceso desde una perspectiva particular y así proporcione información parcial sobre el proceso que se llevaran a cabo. Éstos modelos generales no son descripciones definitivas de los procesos del software (aplicación móvil), más bien son abstracciones de los procesos que se pueden utilizar para el desarrollo del software (aplicación móvil). Puede pensarse en ellos como marcos de trabajo del proceso y que pueden ser adaptados para crear procesos más específicos que se adaptarán al desarrollo de nuestro proyecto según a sus necesidades, el tiempo, el presupuesto, el diseño y todo esto englobado según las características que nos marque **Antojitos Mexicanos “El porvenir”** nuestro cliente.

Modelo lineal secuencial

El modelo lineal secuencial es un proceso de desarrollo secuencial, en el que el desarrollo de software (aplicación móvil) se concibe como un conjunto de etapas que se ejecutan una tras otra. El modelo lineal secuencial de desarrollo en se originó en la

industria y la construcción, donde los cambios a posteriori son caros y difíciles de implementar.

Ingeniería del Sistema. Análisis de las características y el comportamiento del sistema del cual el software va a formar parte. Para un sistema nuevo: Se debe analizar cuáles son los requisitos funciones del sistema, y luego asignar un subconjunto de estos requisitos y funciones al software.

Para un sistema ya existente: se debe analizar el funcionamiento de la organización y sus operaciones y se asigna al software aquellas funciones que se van a automatizar.

Está formado por diagramas y por descripciones en lenguaje natural.

Análisis. Se debe comprender cuáles son los datos que se van a manejar, cuál va a ser la función que tiene que cumplir el software, cuáles son las interfaces requeridas y cuál es el rendimiento y otros requisitos no funcionales que se esperan lograr.

Los requisitos, tanto del sistema como del software deben documentarse y revisarse con el cliente. Como resultado de la fase de análisis, se obtiene la especificación de requisitos del software.

También está formado por diagramas y descripciones en lenguaje natural.

Diseño. El diseño se aplica a cuatro características distintas del software: la estructura de los datos, la arquitectura de las aplicaciones, la estructura interna de los programas y las interfaces.

El diseño es el proceso que traduce los requisitos en una representación del software de forma que pueda conocerse la arquitectura, funcionalidad e incluso la calidad del mismo antes de comenzar la codificación.

En el diseño, los requisitos del software se traducen a una serie de diagramas que representan la estructura del sistema software, de sus datos, de sus programas y de sus interfaces.

Codificación. Consiste en la traducción del diseño a un formato que sea comprensible para la máquina. Si el diseño es lo suficientemente detallado, la codificación es relativamente sencilla, y puede hacerse de forma automática, usando generadores de código.

Se traducen los diagramas de diseño a un lenguaje fuente, que luego se traduce, se compila, para obtener un programa ejecutable.

Prueba. El objetivo es comprobar que no se hayan producido errores en alguna de las fases anteriores, especialmente en la codificación. Se deben probar todas las sentencias, y todos los módulos que forman parte del sistema.

Utilización. El software se entrega al cliente y comienza la vida útil del mismo.

Mantenimiento. El software sufrirá cambios a lo largo de su vida útil. Estos cambios pueden ser debidos a tres causas:

- Que, durante la utilización, el cliente detecte errores en el software: los errores latentes.
- Que se produzcan cambios en alguno de los componentes del sistema.
- Que el cliente requiera modificaciones funcionales no contempladas en el proyecto.

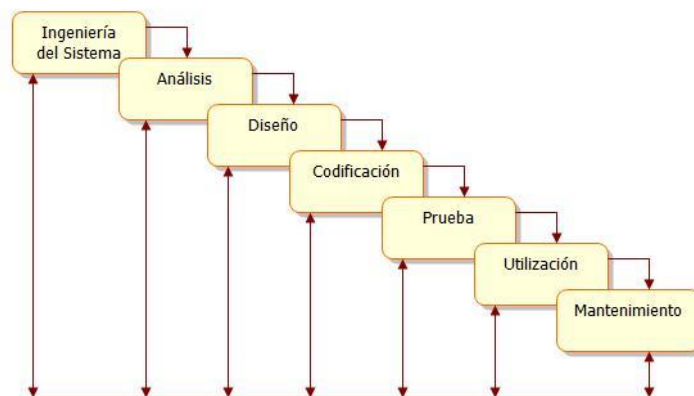


Imagen 1. Esquema modelo Lineal Secuencial

Departamento: Análisis.

Metodología

Prototipado

El Modelo de prototipos, en Ingeniería de software, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos. Durante el proyecto lo veremos aplicado en la aplicación Web Figma donde se llevara a cabo la todo el prototipado de nuestra aplicación móvil.

Etapas

- Recolección y refinamiento de requisitos.
- Modelado, diseño rápido.
- Construcción del Prototipo.
- Desarrollo, evaluación del prototipo por el cliente.
- Refinamiento del prototipo.
- Producto de Ingeniería.
- El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción de un prototipo, el cual es evaluado por el cliente para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará.

Ventajas

- No modifica el flujo del ciclo de vida.
- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.
- Reduce costo y aumenta la probabilidad de éxito.
- Exige disponer de las herramientas adecuadas.

Desventajas.

- Debido a que el usuario ve que el prototipo funciona piensa que este es el producto terminado y no entienden que recién se va a desarrollar el software.
- El desarrollador puede caer en la tentación de ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y mantenimiento que tiene con el cliente.

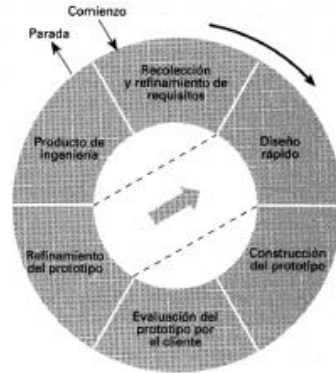


Imagen 2. Esquema modelo Prototipado

Modelo en Espiral

El modelo en espiral es una combinación entre el modelo lineal o de cascada y el modelo iterativo o basado en prototipos que habíamos mencionado anteriormente. Se utiliza con éxito en proyectos donde el coste de un fallo es un gran riesgo, de ahí que su principal aportación sea considerar la gestión de esos riesgos, algo que en los modelos anteriores ni siquiera se menciona.

Las fases por las que pasa cada ciclo de la espiral son:

- Planificación. Se determinan los objetivos y el alcance del ciclo que comienza, tras un necesario ejercicio de investigación. Con cada iteración, se irá incrementando el tamaño de software entregado y la funcionalidad cubierta.
- Análisis de Riesgo. Se evalúa todo aquello que pueda afectar al proyecto según el estado en que se encuentre y su grado de avance. Para ello, se diseñarán los prototipos que deberán ser validados en el ciclo.
- Implementación. Se desarrolla y valida el software según el alcance acordado, el cual está íntimamente relacionado y condicionado con el análisis de riesgos anterior.
- Evaluación. Antes de proceder a realizar otra vuelta en la espiral, se debe prestar atención a lo que sucedió en la vuelta anterior. Se debe analizar en detalle si los riesgos detectados anteriormente ya tuvieron solución. Básicamente, esta fase servirá para

determinar el avance del proyecto y dar pistas de hacia dónde debe enfocarse la próxima iteración.

Ventajas

- Puede adaptarse y aplicarse a lo largo de la vida del software de computadora.
- Es un enfoque realista del desarrollo de sistemas y de software a gran escala.
- Como el software evoluciona, a medida que progresa el proceso el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.

Desventajas

- Puede resultar difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable.
- Requiere una considerable habilidad para la evaluación del riesgo.
- No se ha utilizado tanto como los paradigmas lineales secuenciales o de construcción de prototipos.

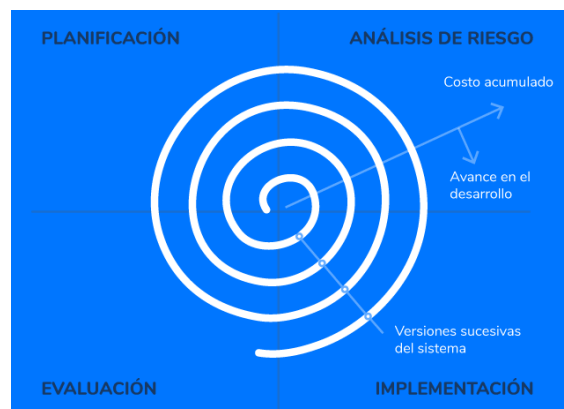


Imagen 3. Esquema modelo Espiral

Departamento: Análisis.

Comparación entre modelos Prototipado y Espiral

CRITERIO	PROTOTIPADO	ESPIRAL
Disponibilidad de recursos	Algunos	Algunos
Complejidad del proyecto	Media	Alta
Entendimiento de requerimientos	Vago	Vago
Tecnología del producto	Vago	Vago
Manejo de la perspectiva de riesgo	Si	Si
Conocimiento del dominio de problemas	Regular	Pobre

Imagen 4. Tabla de comparación de metodologías

Análisis de factibilidades

Entrevista para punto de venta en el local Antojitos Mexicanos “El porvenir”.

1. ¿Posee Computadora e Impresora?

No, son muy estorbosas y corren el riesgo de llenarse de grasa en el local.

2. ¿Posee Teléfonos Móviles?

Si, actualmente por este medio tomamos pedidos de llamadas telefónicas.

3. ¿Cuántos trabajadores tiene su local?

Un mesero, mi esposa y yo que somos los dueños.

4. Al no tener un sistema ¿Cómo actualmente registra sus ventas?

En pedazos de papel destrasa tomamos las órdenes, tanto yo como mi mesero y de igual forma ahí plasmamos la cuenta de lo consumido por el cliente.

5. ¿Qué le gustaría que le reflejara su sistema en cuanto a ventas diarias?

En primera que me dé un corte de cuanto vendo al día por que en la actualidad mi esposa tiene que sumar todos los papelitos generados para saber un total, todos los pedidos con un folio y un resultado de que producto vendí más.

6. ¿Cuál sería la principal causa por la cual implementar un sistema en su negocio?

Una mejor contabilidad de lo que vendo contra lo que invierto, y una mejor atención para mis clientes al generarle con rapidez primero su pedido y próximamente su cuenta, así como no tener errores al darles su cambio.

7. ¿Cuántas categorías de antojitos tiene? ¿Cuáles son?

6. Quesadillas, Pambazos, Tostadas, Sopes, Gorditas y Refrescos.

8. ¿Cuáles son los ingredientes adicionales para cada uno de estos?

Lechuga, Quesillo, Queso Rayado, Crema, Frijoles, Salsa.

9. ¿Qué le parece más atractivo el texto o las imágenes?

Las imágenes, aunque el texto complementaria que es y cuánto cuesta.

10. ¿Cómo sería el pago de sus productos?

Siempre en efectivo ya que no contamos con terminal de tarjetas aun.

11. ¿Cuál es el problema más habitual con sus clientes?

Algunos tienen duda cuando se les da el total de su cuenta.

Departamento: Análisis.

Diagrama de actividades

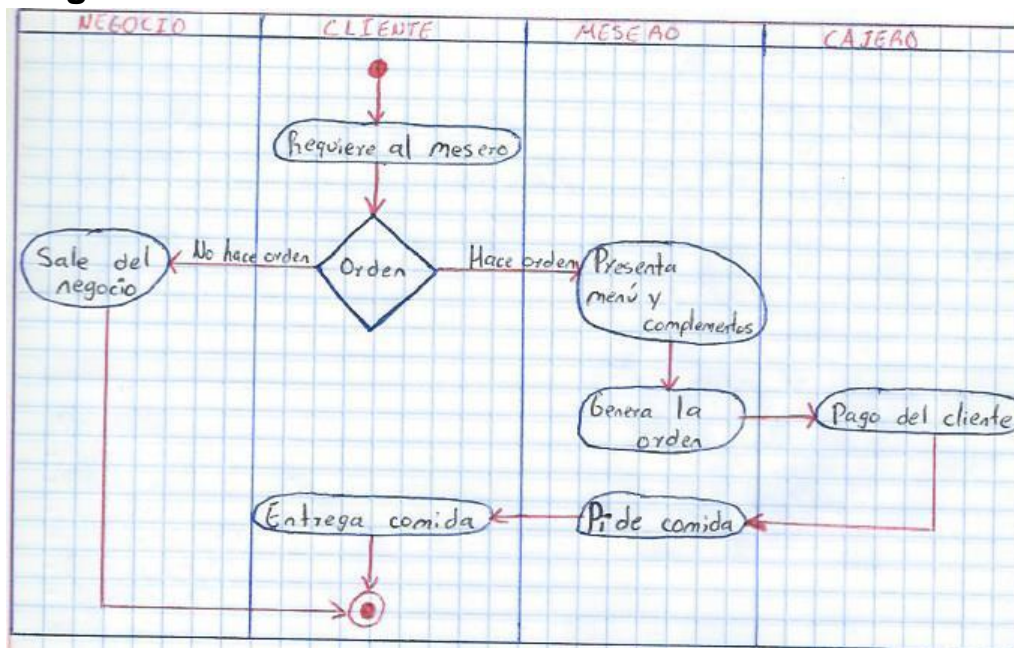


Imagen 5. Diagrama de actividades

Diagrama de flujo de datos

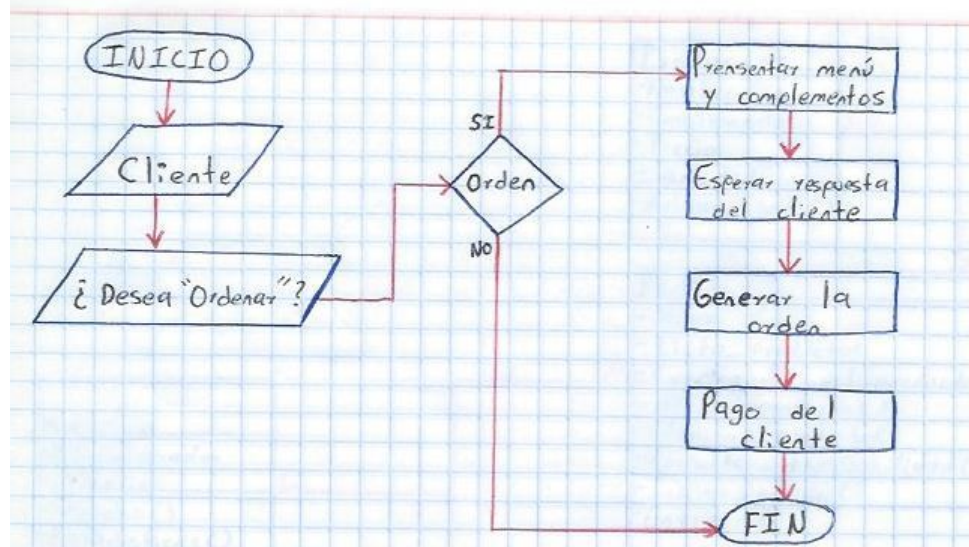


Imagen 6. Diagrama de flujo de datos

Diagrama de clases

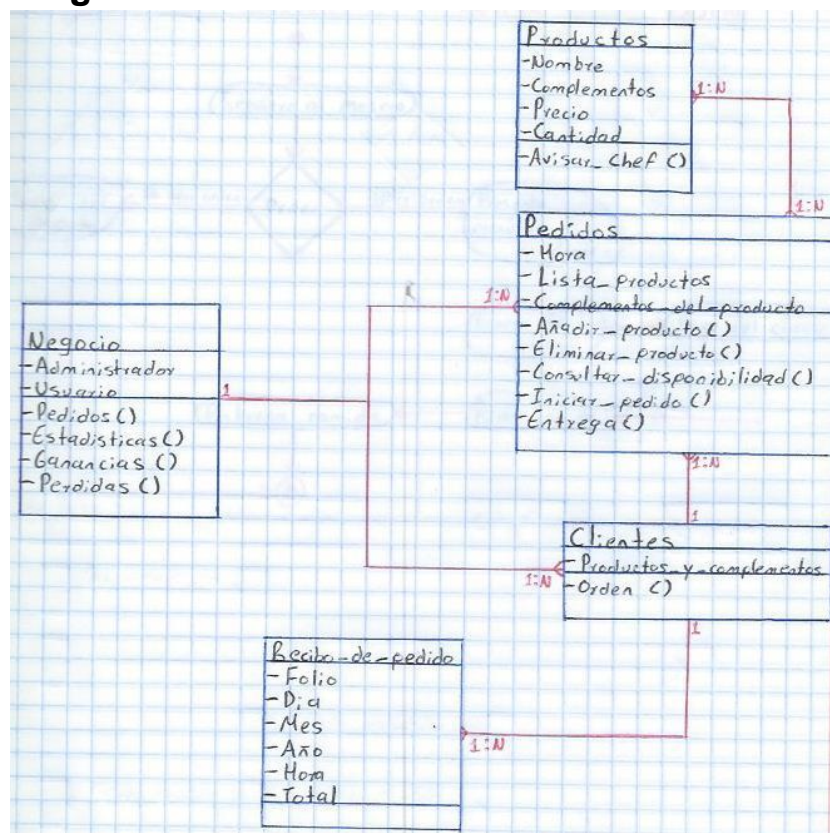


Imagen 7. Diagrama de clases

Diagrama de secuencia

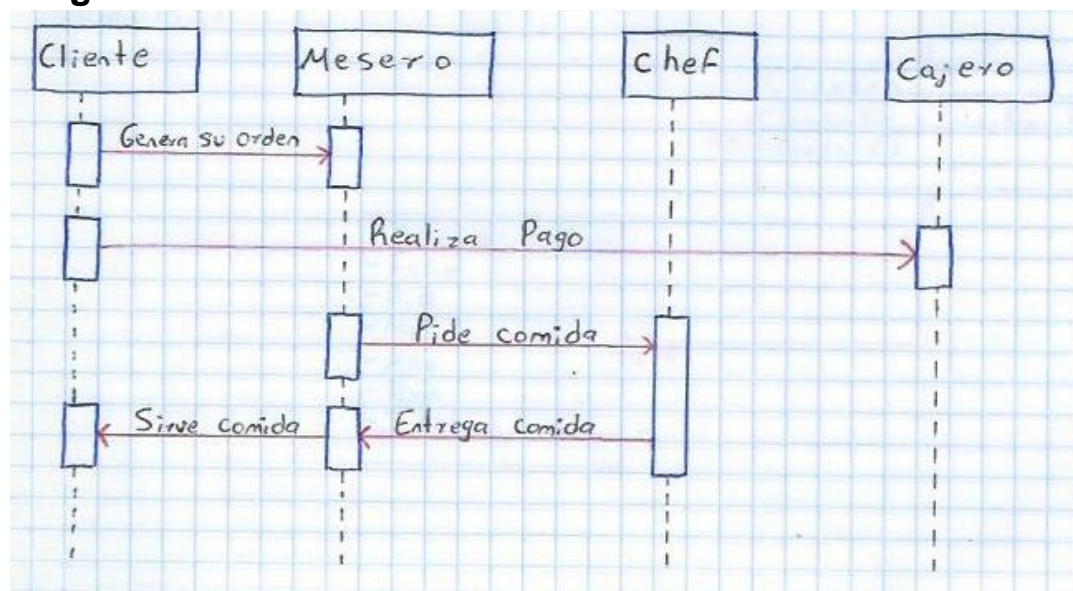


Imagen 8. Diagrama de secuencia

Diccionarios de datos

Campo	Tamaño	Tipo de Dato	Descripción
n_Administrador	5	Carácter	Nombre del administrador
n_Usuario	5	Carácter	Nombre del usuario
c_Administrador	4	Carácter	Contraseña del administrador
c_Usuario	4	Carácter	Contraseña del usuario
Estadísticas	5	Número	Número de pedidos atendidos en día,mes y año
Ganancias	5	Número	Capital de ganancias
Perdidas	5	Número	Capital de pérdidas
Folio	10	Carácter	Número único de orden atendida
Fecha_Hora	1	Datetime	Fecha y hora de atención al pedido
Total	5	Número	Costo total de los productos de una orden
n_Producto	15	Carácter	Nombre del producto
Complementos_del_Produ	15	Carácter	Complementos extra a tu producto lechuga por ejemplo
Precio_uProducto	2	Número	Precio unitario del producto
Cantidad	3	Número	Marcador para saber piezas de cada producto
Campos clave:			
n_Administrador, n_Usuario, c_Administrador, c_Usuario			

6. Estructura lógicas de la información

Diseño

El diseño juega un papel importante en el desarrollo dentro de nuestro proyecto, el cual nos permite producir un modelo del sistema o producto de que se va a construir, el mismo que forma una especie de plan de la solución de la aplicación. Este modelo puede evaluarse en relación con su calidad y mejorarse antes de generar código, de realizar pruebas y de que los usuarios finales se vean involucrados a gran escala. El diseño es el sitio en el que se establece la calidad del software.

Para el proyecto en primera instancia realizamos en clase un bosquejo a mano simulando las pantallas de nuestra app esto con el fin de que si teníamos algún problema en botones o secuencia fuera más fácil de corregir, ya que no tenía un diseño formal de colores dentro de algún software de maquetado.

En segunda instancia dimos paso al software especializado en maquetado el cual es **Figma** donde plasmamos ya todo nuestro bosquejo con los diferentes objetos que empleamos en el mismo,

en este ya utilizamos una paleta de colores descargada de una página reconocida en formato png, esto con la meta de llevar a cabo una buena combinación de colores distinguiendo el mejor a través de 3 combinaciones de los mismos, hechos por medio del login de nuestra aplicación y aplicando la combinación de colores a todas nuestras pantallas que conforman la app.

Diseño de la arquitectura del sistema



Imagen 9. Manual de usuario



Imagen 10. Manual de administrador

Prototipos de interfaces

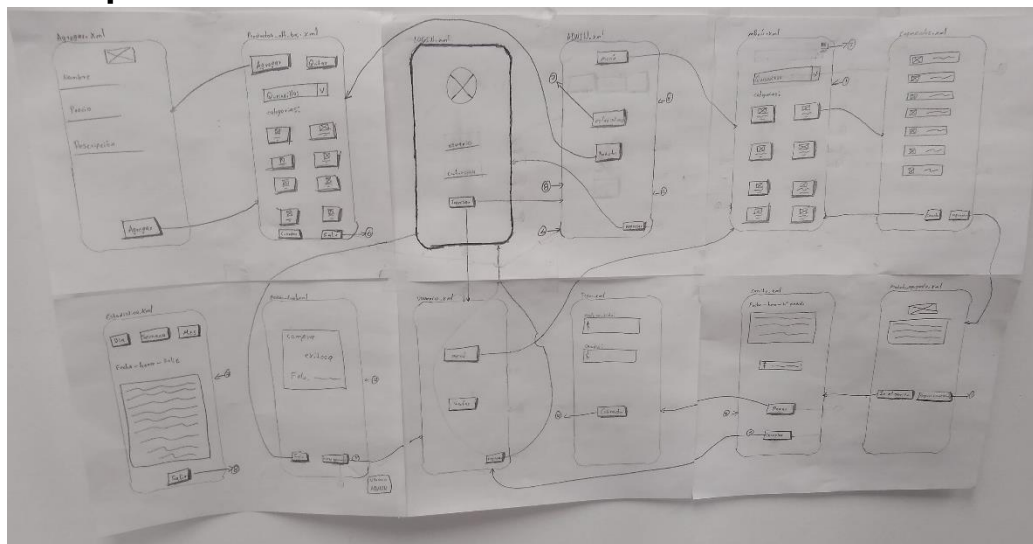


Imagen 11. Bosquejo hecho en clase

7. Componentes y desarrollo SW

Programación

El término Activity (actividad) es de los primeros que debemos entender a la perfección en Android, ya que es de lo más básico y se usa muchísimo en las aplicaciones.

Podemos decir que todas las pantallas de una aplicación son una “activity”. Más adelante vamos a ver que existen algunas variaciones, pero por ahora digamos que todas lo son. Es decir, que, si una aplicación tiene cinco pantallas, tiene 5 “Actividades” o activities.

Las activities están conformadas por dos partes: la parte lógica y la parte gráfica.

La parte lógica es un archivo .java que es la clase que se crea para poder manipular, interactuar y colocar el código de esa actividad.

La parte gráfica es un XML que tiene todos los elementos que estamos viendo de una pantalla declarados con etiquetas parecidas a las del HTML, es decir, que el diseño de una aplicación en Android se hace similar a una página web; XML es un primo de HTML.

Resumiendo, una actividad está conformada por la parte lógica (un archivo Java) y la parte gráfica (un archivo XML).

Adentrando más en el tema, ya sabemos que tenemos un archivo .java, esto quiere decir que tenemos una clase principal, al ser una actividad extiende de la clase Activity (por eso el nombre) que nos proporciona Android para crear actividades con sus métodos asignados.

En Android, la habilidad de enviar mensajes es posible por el objeto Intent. Con la ayuda de los intents, los componentes de Android pueden solicitar funcionalidad de otros componentes Android. Cuando abres la aplicación Instagram en tu teléfono y la usas para tomar una foto, acabas de hacer uso de un intent. Los Intents también ayudan a comunicar entre partes de una app; el movimiento de una pantalla (actividad) a otra es posible mediante intents.

Míralo de esta forma: todos los componentes (aplicaciones y pantallas) del dispositivo Android están aisladas. La única manera de comunicarse entre ellas es a través de intents.

Como se mencionó anteriormente se utilizará en cada pantalla un intent, para pasar de pantalla en pantalla lo cual el código de cada botón nos llevara a una nueva pantalla el código es el siguiente:

```
Boton.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        Intent intent = new Intent(v.getContext(),  
MainActivy.class);  
        startActivityForResult(intent, 0);  
    }  
});
```

8. Gestión de riesgos

Los objetivos de la gestión de riesgos son identificar, dirigir y eliminar las fuentes de riesgo antes de que empiecen a afectar a la finalización satisfactoria de un proyecto software.

El riesgo siempre implica dos características:

-Incertidumbre. El acontecimiento que caracteriza al riesgo puede o no puede ocurrir.

-Pérdida. Si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas.

Para cuantificar el nivel de incertidumbre y el grado de pérdidas asociado con cada riesgo se consideran diferentes categorías de riesgos:

-Riesgos del proyecto. Afectan a la planificación temporal y al coste del proyecto. Identifican problemas potenciales de presupuesto, calendario, personal, recursos.

-Riesgos técnicos. Amenazan la calidad y la planificación temporal del software que hay que producir. Identifican posibles problemas de diseño, implementación, interfaz, verificación y mantenimiento.

-Riesgos del negocio. Amenazan la viabilidad del software. Los principales riesgos de negocio son:

- *Riesgo de mercado.

- *Riesgo estratégico.

- *Riesgo de ventas.

- *Riesgo de dirección.

- *Riesgo de presupuesto.

Se puede hacer otra categorización de los riesgos en función de su facilidad de detección:

-Riesgos conocidos. Son aquellos que se pueden predecir después de una evaluación del plan del proyecto, del entorno técnico y otras fuentes de información fiables.

-Riesgos predecibles. Se extrapolan de la experiencia de proyectos anteriores.

-Riesgos impredecibles. Pueden ocurrir, pero es extremadamente difícil identificarlos por adelantado.

La gestión continuada de los riesgos permite aumentar su eficiencia:

- Evaluar continuamente lo que pueda ir mal.
- Determinar qué riesgos son importantes o implementar estrategias para resolverlos.
- Asegurar la eficacia de las estrategias.

9.Verificación de la calidad

Pruebas

Las pruebas llevadas a cabo fueron únicamente de **caja blanca** ya que solo tenemos funcionamiento en botones para cambio de pantallas, estos adecuados a ingresar en tan solo un toque de nuestro dedo en los botones marcados como **siguiente o ingresar y regresar o cancelar**.

Normativas y protocolos

Entre las principales normativas y protocolos podemos hacer mención de tres, las cuales hacen mención de las buenas prácticas para desarrollar software.

ISO 9001. Es una norma ISO internacional elaborada por la Organización Internacional para la Estandarización (ISO) que se aplica a los Sistemas de Gestión de Calidad de organizaciones públicas y privadas, independientemente de su tamaño o actividad empresarial. Se trata de un método de trabajo excelente para la mejora de la calidad de los productos y servicios, así como de la satisfacción del cliente.

El sistema de gestión de calidad se basa en la norma ISO 9001, las empresas se interesan por obtener esta certificación para garantizar a sus clientes la mejora de sus productos o servicios y

estos a su vez prefieren empresas comprometidas con la calidad. Por lo tanto, las normas como la ISO 9001 se convierten en una ventaja competitiva para las organizaciones.

ISO/IEC 15504. El Estándar internacional ISO/IEC 15504 denominado como Software Process Improvement Capability Determination (Determinación de la Capacidad de Mejora del Proceso de Software), nos propone un modelo para la evaluación de la capacidad en los procesos de desarrollo de productos Software.

ISO 33000. La familia de normas ISO/IEC 33000 proporciona un marco de trabajo coherente para la evaluación de procesos software que sustituye las diferentes partes de la norma ISO/IEC 15504.

El propósito de la serie de estándares ISO/IEC 33000 es proporcionar un enfoque estructurado para la evaluación de procesos, permitiendo a las organizaciones lograr distintos objetivos:

- Comprender el estado de sus propios procesos buscando la mejora de los mismos.
- Determinar la idoneidad de sus propios procesos para un requerimiento en particular o para un conjunto de requerimientos.
- Determinar la idoneidad de los procesos de otra organización para un contrato específico o para un conjunto de contratos.

10. Entregables del proyecto

- Maquetado en la aplicación Figma a través del link donde se desarrolló.
- Carpeta completa del proyecto de Android Studio.
- Manual de usuario ilustrado por pantallas a interactuar dentro de la aplicación.

11. Catálogo de imágenes

- Imagen 1. Esquema modelo Lineal Secuencial [10]
- Imagen 2. Esquema modelo Prototipado [12]
- Imagen 3. Esquema modelo Espiral [13]
- Imagen 4. Tabla de comparación de metodologías [14]
- Imagen 5. Diagrama de actividades [16]
- Imagen 6. Diagrama de flujo de datos [16]
- Imagen 7. Diagrama de clases [17]
- Imagen 8. Diagrama de secuencia [17]
- Imagen 9. Manual de usuario [20]
- Imagen 10. Manual de administrador [21]
- Imagen 11. Bosquejo hecho en clase [22]

12. Referencias

<https://ingsoftwarejefer.webcindario.com/unidad-1/modelos-de-proceso-del-software/modelo-lineal-secuencial.html>

<https://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm#:~:text=2.11.2%20Modelos%20para%20el%20desarrollo%20de%20software&text=Considera%20las%20actividades%20fundamentales%20del,implementaci%C3%B3n%20C%20las%20pruebas%20C%20etc%C3%A9tera.>

<https://alexbalmoregomezcornejo.blogspot.com/2012/05/modelo-de-prototipos-y-modelo-en.html>

<https://aspgems.com/metodologia-de-desarrollo-de-software-iii-modelo-en-espiral/>

https://www.lucidchart.com/pages/es/ejemplos-y-plantillas-de-diagramas-uml#section_4

https://www.ecured.cu/Modelo_de_prototipos

https://www.ecured.cu/Dise%C3%B1o_de_software#:~:text=Dise%C3%B1o%20de%20software.,que%20resulta%20de%20este%20proceso.

<https://sites.google.com/site/gestiondeproyectossoftware/unidad-3-planificacion-de-proyecto/3-5-1-tipos-de-riesgos>

<https://www.isotools.org/normas/calidad/iso-9001/>

<https://www.normas-iso.com/iso-iec-15504-spice/>

<https://www.iso33000.es/>