



Análisis Y Diseño Orientado a Objetos

Informe de Proyecto Aplicando el Proceso Unificado (UP): Caso de Estudio “Comedor UAZ Siglo XXI”

GRUPO 3B

INTEGRANTES:

Luis Diego Morales

Berenice Nayeli Madrid Rodarte

Jimena Sofía Sánchez

FASE DE COMIENZO (INCEPTION)

Objetivo: Determinar si el proyecto es viable y establecer una visión general.

Visión y Caso de Negocio

El presente documento detalla la planificación y el alcance del desarrollo de un sistema web integral, diseñado específicamente para optimizar la gestión del comedor universitario ubicado en el Campus Siglo XXI de la Universidad Autónoma de Zacatecas.

El presente documento tiene como objetivo principal detallar de manera exhaustiva la planificación, el alcance y los objetivos de un sistema web integral. Este sistema está diseñado específicamente para optimizar y modernizar la gestión operativa del comedor universitario, una pieza clave en la infraestructura de servicios para estudiantes y personal. El comedor se encuentra estratégicamente ubicado en el Campus Siglo XXI de la Universidad Autónoma de Zacatecas (UAZ), un centro educativo de vital importancia en la región.

La necesidad de este desarrollo surge de la identificación de áreas de mejora en los procesos actuales del comedor, buscando la eficiencia, la reducción de costos operativos y la mejora significativa de la experiencia del usuario, tanto para los comensales como para el personal administrativo y de servicio. A través de este sistema web, se pretende abordar diversas funcionalidades clave, incluyendo, pero no limitándose a: la gestión de inventarios de alimentos, la planificación y programación de menús, el seguimiento de la demanda y el consumo, la administración de pagos y suscripciones de estudiantes, la generación de reportes detallados para la toma de decisiones, y la implementación de un sistema de retroalimentación que permita una mejora continua del servicio.

Este proyecto representa un esfuerzo por integrar tecnología de vanguardia en la gestión de servicios universitarios, con el fin de proporcionar un entorno más eficiente, transparente y satisfactorio para toda la comunidad del Campus Siglo XXI. Se espera que la implementación exitosa de este sistema no solo optimice las operaciones diarias del comedor, sino que también contribuya a una mejor asignación de recursos, a una reducción del desperdicio de alimentos y a una mayor satisfacción general de los usuarios.

Este proyecto representa un esfuerzo significativo para modernizar la infraestructura de servicios universitarios, mejorando la calidad de vida de la comunidad estudiantil y

optimizando la operación del comedor. Esto es valioso porque impacta directamente en el bienestar de los estudiantes y en la eficiencia de los recursos universitarios.

Objetivos del Proyecto:

Este proyecto se concibe como una solución tecnológica avanzada para abordar y resolver diversas problemáticas que afectan directamente la experiencia de los estudiantes en el comedor universitario. Los objetivos principales incluyen:

- ❖ **Mejora de la Disponibilidad de Alimentos:** Implementar funcionalidades que permitan a los estudiantes conocer en tiempo real el menú del día, la disponibilidad de los platillos y las opciones nutritivas, facilitando así una elección informada y reduciendo la incertidumbre sobre la oferta alimentaria.
- ❖ **Mejora de la Comunicación de Información Relevante:** Establecer un canal de comunicación directo y eficiente para difundir avisos importantes, promociones, encuestas de satisfacción, cambios en el servicio o cualquier otra información de interés para la comunidad universitaria, garantizando que los estudiantes estén siempre actualizados.
- ❖ **Gestión Eficiente de Recursos:** Proporcionar herramientas administrativas para el personal del comedor que permitan una mejor gestión de inventarios, planificación de menús, seguimiento de la demanda y análisis de preferencias de los estudiantes, lo que contribuirá a una operación más eficiente y sostenible.
- ❖ **Optimización de Tiempos de Espera:** Especificar la demanda de los platillos, con el fin de minimizar las filas y los tiempos de espera durante las horas pico, mejorando la eficiencia del servicio.

Enfoque Metodológico:

Para la conceptualización, análisis, diseño y desarrollo de este sistema, se adoptará un **enfoque orientado a objetos (OO)**. Esta metodología se considera la más adecuada debido a su capacidad para modelar sistemas complejos de manera intuitiva y modular, lo que facilita la comprensión de las interacciones entre los diferentes componentes del sistema (como usuarios, menús, pedidos, pagos, etc.). El enfoque orientado a objetos permitirá:

- **Modularidad:** Descomponer el sistema en objetos independientes y reutilizables, lo que simplificará el desarrollo, las pruebas y el mantenimiento.
- **Flexibilidad:** Adaptar el sistema a futuras necesidades y cambios, gracias a la encapsulación y la herencia, que permiten extender funcionalidades sin afectar el código existente.
- **Escalabilidad:** Diseñar una arquitectura robusta que pueda manejar un creciente número de usuarios y transacciones, asegurando un rendimiento óptimo a medida que el servicio se expanda.
- **Colaboración:** Facilitar el trabajo en equipo de los desarrolladores, ya que cada componente puede ser diseñado y probado de forma independiente.

Reglas de negocio

Modelo de Casos de Uso

Actores Identificados

Actor	Descripción
Estudiante	Usuario principal del comedor. Realiza pagos, consulta menús y proporciona retroalimentación.
Administrador del Comedor	Supervisa operaciones, gestiona inventarios, menús y reportes.
Personal de Cocina	Consulta los menús y controla el uso de ingredientes.
Cajero o Personal de Cobro	Administra los pagos, suscripciones y registros de asistencia.
Sistema Bancario	Actor externo que valida y procesa los pagos en línea.
Proveedor	Entrega insumos alimenticios y actualiza el estado de los pedidos.

Metas Principales de los Usuarios

Actor	Meta
Estudiante	Acceder fácilmente a menús, realizar pagos en línea y consultar su historial de consumo.
Administrador	Tener control total sobre inventarios, menús y reportes para optimizar recursos.
Personal de Cocina	Preparar los alimentos con base en menús actualizados y control de ingredientes disponibles.
Cajero o Personal de Cobro	Registrar pagos o asistencias de comensales de forma rápida y sin errores.
Proveedor	Recibir pedidos de manera eficiente y actualizarlos en línea.

Casos de Uso Principales (10 % más importantes)

Estos representan los **casos más críticos o de mayor valor/riesgo** para el éxito del sistema.

# Caso de Uso	Actores Principales	Descripción Breve (en formato casual)	Valor / Riesgo
1.- Consultar el menú del dia	Estudiante / Cocinero	El usuario puede ver el menú actualizado con información nutricional.	Alto valor (satisfacción del usuario)
2.- Generar reportes de consumo y costos	Administrador	El sistema produce reportes sobre ventas, consumo y desperdicio para mejorar decisiones.	Alto valor estratégico
3.- Gestionar inventario de alimentos	Administrador / Cocinero	Permite crear y modificar menús según disponibilidad de ingredientes.	Alto riesgo (impacta la operación diaria)
4.- Planificar y actualizar menús semanales	Administrador	Permite crear y modificar menús según disponibilidad de ingredientes.	Alto riesgo (impacta toda la cadena operativa)
5.- Registrar retroalimentación del comensal	Estudiante	Los usuarios pueden dejar comentarios sobre la calidad del servicio o la comida.	Valor medio-alto (mejora continua)

Nota: esto se modificó posteriormente después de lo acordado en una de las reuniones realizadas con la responsable administrativa del comedor Alejandra Elizabeth Klein Hernandez y el personal del comedor

Requisitos No Funcionales

Requisitos No Funcionales - Basados en el modelo FURPS+

Categoría	Requerimientos No Funcionales Críticos
Fiabilidad (Reliability)	<p>El sistema deberá tener una disponibilidad mínima del 99% durante las horas de servicio del comedor.</p> <p>La información de disponibilidad de platillos y aglomeración deberá actualizarse automáticamente al menos cada 30 segundos.</p> <p>En caso de pérdida de conexión, el sistema deberá recuperar los datos automáticamente sin necesidad de reiniciar sesión.</p> <p>Se realizarán copias de seguridad automáticas de la base de datos cada 24 horas.</p>
Usabilidad (Usability)	<p>La interfaz deberá ser intuitiva, visual y accesible para usuarios con conocimientos básicos de tecnología.</p> <p>Los menús y datos nutricionales deberán ser fáciles de leer y navegar, utilizando íconos y colores distintivos.</p> <p>Deberá incluir una sección de ayuda contextual con instrucciones claras.</p> <p>El sistema será completamente responsive, adaptable a smartphones, tablets y computadoras.</p> <p>El sistema ofrecerá retroalimentación inmediata ante acciones del usuario (por ejemplo, confirmaciones o alertas).</p>
Rendimiento (Performance)	<p>El tiempo de carga de las páginas principales (menú del día, disponibilidad y avisos) deberá ser menor a 3 segundos.</p>

	<p>El sistema deberá soportar hasta 1000 usuarios simultáneos sin pérdida perceptible de rendimiento.</p> <p>Las notificaciones o alertas deberán enviarse al usuario en menos de 5 segundos tras detectarse un cambio en el sistema (por ejemplo, agotamiento de un plátano).</p>
Soportabilidad (Supportability)	<p>El sistema deberá estar diseñado con arquitectura modular, facilitando el mantenimiento y futuras actualizaciones.</p> <p>Deberá permitir configurar parámetros (horarios, menús, avisos) desde un panel administrativo, sin requerir cambios de código.</p> <p>El sistema deberá ser compatible con navegadores modernos (Chrome, Edge, Firefox, Safari).</p> <p>La documentación técnica y de usuario deberá mantenerse actualizada y disponible para el personal del comedor.</p>
Seguridad (FURPS+ Extra)	<p>Todo acceso al sistema deberá requerir autenticación con credenciales seguras(usuario/contraseña encriptada).</p> <p>Los roles de usuario estarán claramente definidos (estudiante, personal, administrador) y tendrán diferentes niveles de acceso.</p> <p>Los datos personales (nombre, correo, etc.) deberán almacenarse conforme a la Ley Federal de Protección de Datos Personales (LFPDPPP).</p> <p>Las acciones administrativas (actualización de menú, inventario o avisos) deberán registrarse en un log de auditoría.</p>
Mantenibilidad / Extensibilidad	<p>El sistema deberá permitir la integración futura con sensores de conteo de personas o sistemas IoT del comedor.</p> <p>El código deberá seguir buenas prácticas de desarrollo orientado a objetos, favoreciendo la extensibilidad y</p>

	<p>reutilización.</p> <p>Se deberá poder agregar nuevos módulos (por ejemplo, “Estadísticas de consumo”) sin afectar la estructura existente.</p>
--	---

Se priorizan los **requisitos de disponibilidad, usabilidad y rendimiento**, pues la experiencia del estudiante depende de información actualizada y rápida.

La **seguridad y la mantenibilidad** son claves para el personal administrativo y técnico.

Lista de Riesgos y Plan de Gestión de Riesgos

El siguiente documento presenta los riesgos identificados en la fase de Inception para el proyecto del Sistema Web de Gestión del Comedor Universitario. Los riesgos se clasifican en técnicos, de negocio y de recursos, con su impacto, probabilidad y estrategias de mitigación correspondientes.

Tipo de Riesgo	Descripción del Riesgo	Impacto Potencial	Probabilidad	Estrategia de Mitigación / Respuesta
Técnico	Fallas en la conexión a Internet o baja velocidad que afecten la actualización en tiempo real de disponibilidad de platillos.	Alta	Media	Implementar almacenamiento local temporal (cache) y sincronización automática cuando se recupere la conexión.
Técnico	Errores en el sistema de notificaciones o lentitud en el envío de alertas.	Media	Media	Optimizar los procesos de actualización y establecer un sistema de monitoreo de rendimiento.
Técnico	Incompatibilidad con algunos navegadores o dispositivos móviles.	Media	Alta	Realizar pruebas de compatibilidad (Chrome, Edge, Firefox, Safari, Android e iOS) antes del despliegue.

Negocio / Alcance	Cambios en los requerimientos por parte de la administración del comedor durante el desarrollo.	Alta	Alta	Definir y validar los requerimientos clave desde la fase de análisis. Mantener comunicación constante con los responsables.
--------------------------	---	------	------	---

Negocio / Aceptación	Baja adopción de la app por parte de los estudiantes.	Alta	Media	Diseñar una interfaz atractiva, sencilla y promover el uso mediante campañas internas en la universidad.
Recursos Humanos	Falta de personal técnico disponible para mantenimiento o soporte.	Alta	Media	Capacitar al personal del comedor y documentar los procedimientos básicos del sistema.
Recursos Tecnológicos	Insuficiencia en los servidores o hosting, provocando lentitud o caídas.	Alta	Alta	Escoger un proveedor de hosting escalable con monitoreo constante del rendimiento.
Seguridad	Accesos no autorizados o manipulación de datos (inventarios o menús).	Alta	Media	Implementar roles, autenticación segura y registros de auditoría para

				cada acción administrativa.
Cronograma	Retrasos en las fases de desarrollo por falta de pruebas o cambios inesperados.	Media	Media	Aplicar una planificación ágil (Scrum o iteraciones cortas) con revisiones periódicas del progreso.
Dependencia Externa	Fallas en APIs o servicios externos usados para notificaciones o análisis de datos.	Media	Baja	Establecer planes alternos (backups locales) y limitar dependencias externas críticas.

Resumen General

Los riesgos más críticos se concentran en los aspectos técnicos (actualización en tiempo real y compatibilidad) y de adopción (uso por parte de los estudiantes). Las estrategias clave se centran en pruebas continuas, documentación, capacitación y comunicación constante entre desarrolladores y el personal del comedor universitario.

Fase II : Elaboración

Evidencia: Minuta de Reunión

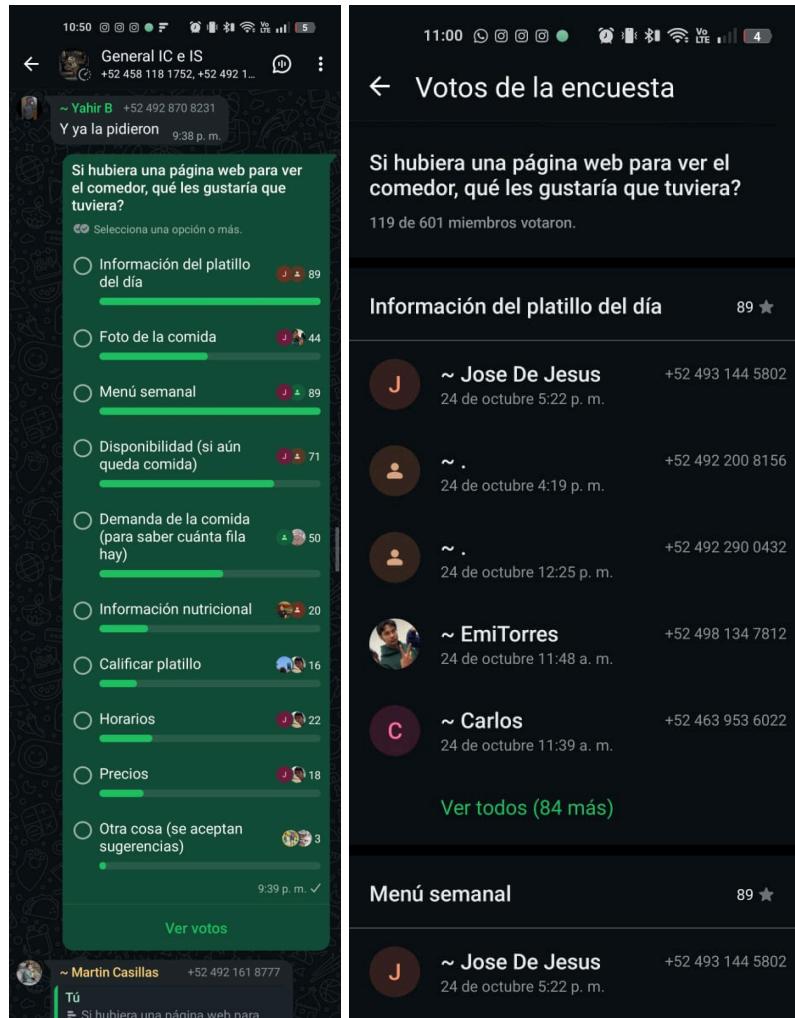
Proyecto: Sistema Web Integral para la Gestión del Comedor Universitario UAZ

Tipo de Reunión: Taller de Requerimientos 1 (Inicio de Fase de Elaboración) **Fecha:** 21 de octubre de 2025.

Tipo de Validación Estudiantil: Encuesta realizada a Estudiantes de la UAZ campus Siglo SXXI

Validación Estudiantil

Se realizó una encuesta a los estudiantes de la UAZ Campus Siglo XXI, de las carreras de Ingeniería de Computación e Ingeniería de Software, con respuestas de 119 estudiantes de ambas carreras. Esta encuesta fue realizada el 23 de octubre. A continuación se adjunta evidencia y resultados de la encuesta:



Se observa que se tiene una mayor aprobación por mostrar información del platillo del día, el menú

semanal, disponibilidad, demanda y foto de la comida, por que mostrar esa información se tomará como prioridad para el desarrollo del sistema.

Minuta de Reunión

8 de octubre del 2025

Asistentes:

- Administración del Comedor
- Ingenieros: Luis Diego Morales, Berenice Nayeli Madrid Rodarte,, Jimena Sofía Sánchez

Agenda

1 - Revisar y confirmar los artefactos de la Fase de Inception.
2 - Definir la priorización de Casos de Uso para la Iteración 1 (Elaboración).
3 - Acordar el primer Caso de Uso a detallar en formato "Completamente Vestido".
4 - Establecer los siguientes pasos para el modelado de análisis
5 - Aclaración y refinamiento del Alcance del Proyecto (Pagos y Reservas)

1. Conforme a la agenda revisar y confirmar los artefactos de la Fase de Inception, se volvió a revisar todo el documento, el equipo llega a un acuerdo que todo queda medianamente claro basándose en la metodología que todo es una Visión general.
2. Hablando la encargada del comedor, la Dra. Alma Carmina Ornelas Pargas nos comenta que no podemos modificar el sistema de pagos de la universidad, a causa de que el sistema de registro de pagos no es perteneciente de la Universidad, sino que es una normativa externa, por lo que se dejó de lado los casos de uso del sistema de pago.
3. Hablando con la Responsable Administrativa Alejandra Elizabeth Klein Hernandez, nos confirma que la lista de las funcionalidades son correctas de acuerdo a lo corregido, nos indica la importancia de la funcionalidad de calificar los platillos por parte de los estudiantes y nos explica qué datos necesitan para dar una idea de la organización de la base de datos.
4. La Responsable Administrativa indica que tampoco se gestionará el inventario, ya que muchas veces se cuentan con proveedores variables.
5. Entre el equipo llegamos a una conclusión donde se modificará uno de los casos de uso, (Sistema de cobro, reserva, cajero) donde se llega a una conclusión que para el lanzamiento no es factible.
6. La Responsable Administrativa Alejandra Elizabeth Klein Hernandez, confirma que la lista de Casos de Uso priorizados por riesgo/valor es la base para iniciar esta fase.
7. También refinamos y aclaramos la idea del Documento Inception referente a (Pagos y

Reservas).

Validación Del personal del Comedor

Asistentes:

- Personal 1 de Servicio del comedor
- Personal 2 de Servicio del comedor
- Ingenieros: Luis Diego Morales, Berenice Nayeli Madrid Rodarte,, Jimena Sofía Sánchez

Agenda

1 - Revisar y confirmar los artefactos de la Fase de Inception.
2 - Definir la priorización de Casos de Uso para la Iteración 1 (Elaboración).

1. Conforme a la agenda revisar y confirmar los artefactos de la Fase de Inception, se volvió a revisar todo el documento, el equipo llega a un acuerdo que todo queda medianamente claro basándose en la metodología que todo es una Visión general.
2. Hablando con el personal nos confirma que las modificaciones realizadas son correctas



SOMOS
ARTE, CIENCIA Y
DESARROLLO
CULTURAL

OS
INGENIERÍA
DE SOFTWARE

Dra. Rosa Adriana Martínez Esquivel
Encargada de la Licenciatura en Nutrición
P r e s e n t e

Asunto: Solicitud de autorización para entrevista y visita al Comedor Universitario

Distinguida Dra. Martínez Esquivel:

Por medio de la presente, me permito solicitar atentamente su autorización para que la alumna Berenice Nayeli Madrid Rodarte, estudiante del Programa de Ingeniería de Software de la Universidad Autónoma de Zacatecas, con correo electrónico 42402159@uaz.edu.mx, pueda sostener una entrevista con el personal del Comedor Universitario e ingresar a las instalaciones de la cocina.

El objetivo de dicha visita es recabar información sobre el proceso operativo y las necesidades del servicio alimentario, con el fin de desarrollar una aplicación de software que contribuya a mejorar la gestión y comunicación del comedor universitario, permitiendo conocer la disponibilidad de platillos, horarios, precios y menú del día.

La información obtenida será utilizada únicamente con fines académicos, como parte de un proyecto de desarrollo de software dentro de la Licenciatura en Ingeniería de Software de esta casa de estudios.

Agradezco de antemano su atención y apoyo a esta iniciativa, quedando a disposición para atender cualquier indicación o requisito adicional necesario para la realización de la entrevista o visita.

Sin más, por el momento, reciba un cordial saludo.

Atentamente,

Dr. Alejandro Mauricio González

Responsable del Programa Académico
Ingeniería de Software – UAZ

Recibí
Carmiña Ornelas
Jo. B. O.
5-V-12

Consorcio de
Universidades
Mexicanas

UNIVERSIDAD AUTÓNOMA DE ZACATECAS "FRANCISCO GARCÍA SALINA"
Carr. Zacatecas – Guadalajara Km. 6 / La Escondida / Ciudad Universitaria Campus Siglo XXI / Código Postal 98160
+52 (402) 925 6600, Extensión 10

2. Decisiones y Acuerdos (Cliente e Ingeniero)

Tema	Discusión / Decisión
Revisión del Alcance - Pagos en Línea	<p>Cliente: Aclaró que el proyecto no debe modificar el sistema de cobro actual. El comedor seguirá usando el cobro en efectivo y el sistema de filas (primero en llegar, primero en ser atendido).</p> <p>Ingeniero: Entendido. Esto es una aclaración crítica que cambia el alcance.</p> <p>Acuerdo: El sistema no gestionará pagos, suscripciones ni reservas. El rol de "Cajero" se enfocará solo en realizar actualizaciones puntuales de los platillos y publicar avisos para los estudiantes.</p>
Re-priorización de la Iteración 1	<p>Cliente: El valor principal para el Estudiante es la información: Consultar menú del día, ver la disponibilidad de platillos en tiempo real.</p> <p>Los platillos que servimos no se repiten tanto como para que sea útil guardar todos los menús además de la semana actual, pero si nos gustaría poder guardar menús con anticipación en caso de no tener tiempo después.</p> <p>Ingeniero: Siguiendo la metodología UP (architecture-centric), propongo que la Iteración 1 se centre en la arquitectura de publicar los platillos de la semana actual y hacer modificaciones en los platillos en curso: De acuerdo. Es la base para que el estudiante pueda ver la información real. Acuerdo: La Iteración 1 de la Fase de Elaboración se centrará en el análisis y diseño de agregar platillos al menú de la semana y editar los mismos.</p>
Primer Caso de Uso a Detallar	<p>Ingeniero: Propongo empezar con el caso de uso Agregar platillo que permita publicar el desayuno y la comida de la semana actual, y el caso de uso Editar Platillo para realizar modificaciones básicas de la información de un platillo existente así como subir una foto del platillo que se está sirviendo en el momento.</p>

3. Próximos Pasos (Plan de Iteración 1)

ACCIÓN (Ingeniero): Generar el borrador del Caso de Uso **UC-02: Publicar platillo** dentro de la semana actual y **UC-03: Editar Platillo**.

ACCIÓN (Ingeniero): Basado en el UC-02, generar el borrador del **Modelo de Dominio**.

ACCIÓN (Cliente): Revisar y validar el Caso de Uso UC-02 y UC-03.

Caso de Uso UC-01: Iniciar Sesión Administrador

Sección	Comentario
---------	------------

Nombre del Caso de Uso	Iniciar Sesión Administrador
Alcance	Sistema Web de publicación del Comedor Universitario UAZ
Nivel	Objetivo del administrador
Actor Principal	Administrador del Comedor
Stakeholders e Intereses	Maestro Joaquin Acevedo Sanchez Alejandra Elizabeth Klein Hernandes (Responsable administrativo)
Precondiciones	El Administrador debe tener una cuenta de administrador en el sistema El sistema debe estar conectado a la base de datos.
Postcondiciones	Exito El Administrador ingresa correctamente a la parte de administrador Fracaso El Sistema bloquea el inicio de sesión durante 30 minutos

FLUJO NORMAL

- 1.- El Administrador selecciona la opción “Ingresar Admin” en la pantalla de inicio.
- 2.- El sistema solicita el email y la contraseña
- 3.- El Administrador llena los campos solicitados.
- 4.- El Administrador da click en “Ingresar”.
- 5.- El Sistema valida el email y la contraseña y redirige a la pantalla principal de administradores.

Extensiones (Flujos Alternativos)

- 5a. El email ingresado no está registrado
1. El Sistema informa al Administrador que el email proporcionado no se encuentra.
 - Regresa al paso 2 del flujo normal.

5b. Contraseña incorrecta

1. El Sistema informa que la contraseña es incorrecta.
 - Regresa al paso 2 del flujo normal.

3a. Recuperar contraseña:

1. El Administrador selecciona “Olvidé mi contraseña”.
2. El Sistema manda un correo electrónico de recuperación de contraseña
 - Regresa al paso 3 del flujo normal

Flujos de excepcion

5c. El Administrador ingresa una contraseña incorrecta 3 veces

1. El Sistema muestra el mensaje “Cuenta bloqueada, inténtelo más tarde”

Caso de Uso UC-02: Publicar platillo

Sección	Comentario
Nombre del Caso de Uso	Publicar Platillo
Alcance	Sistema Web de publicación del Comedor Universitario UAZ
Nivel	Objetivo del administrador
Actor Principal	Administrador del Comedor
Stakeholders e Intereses	Maestro Joaquin Acevedo Sanchez Alejandra Elizabeth Klein Hernandes (Responsable administrativo)
Precondiciones	<p>El administrador debe haber iniciado sesión en el sistema con una cuenta autorizada.</p> <p>El sistema debe estar conectado a la base de datos.</p> <p>Debe existir una estructura definida para mostrar la información en la página pública.</p>
Postcondiciones	<p>Exito La información del comedor se actualiza y se muestra correctamente en la interfaz visible para los estudiantes.</p> <p>El sistema guarda el platillo en la base de datos.</p> <p>Fracaso El sistema no guarda el platillo en la base de datos.</p>

FLUJO NORMAL

- 1.- El Administrador selecciona la opción “**Agregar Platillo**”.
- 2.- El sistema muestra un formulario con los campos para ingresar toda la información: platillo principal, complementos, fecha, tipo (desayuno / comida) disponibilidad, demanda, información nutricional.
- 3.- El Administrador llena los campos solicitados.
- 4.- El Administrador confirma la publicación presionando el botón “**Guardar**”.
- 5.- El Sistema valida los datos ingresados y muestra el mensaje “Agregado con éxito” y regresa a la pantalla principal.

Extensiones (Flujos Alternativos)

*a. En cualquier momento del flujo normal, el Administrador presiona “Cancelar”:

1. El Sistema pregunta si está seguro de cancelar
 - 1a.- El Administrador presiona “Aceptar”.
 - El Sistema descarta los cambios y regresa a la pantalla principal. Termina el caso de uso.
 - 1b.- El Administrador presiona “Cancelar”.
 - Regresa al paso 2 del flujo normal

Flujos de Excepción

5b. Error al guardar la información

1. El sistema notifica “Error al guardar los datos. Intente nuevamente más tarde”.

*a. En cualquier momento el Administrador sale inesperadamente del sistema o la conexión falla

1. El Sistema descarta los cambios y devuelve a la pantalla principal cuando se restablezca la conexión

Caso de Uso UC-03: Editar Platillo

Sección	Comentario
---------	------------

Nombre del Caso de Uso	Editar Platillo
Alcance	Sistema Web de publicación del Comedor Universitario UAZ
Nivel	Objetivo del administrador
Actor Principal	Administrador del Comedor
Stakeholders e Intereses	Maestro Joaquin Acevedo Sanchez Alejandra Elizabeth Klein Hernandes (Responsable administrativo)
Precondiciones	<p>El administrador debe haber iniciado sesión en el sistema con una cuenta autorizada.</p> <p>El sistema debe estar conectado a la base de datos.</p> <p>Hay al menos un platillo creado</p>
Postcondiciones	<p>Éxito La información del comedor se actualiza y se muestra correctamente en la interfaz visible para los estudiantes.</p> <p>El Sistema guarda los cambios realizados por el administrador en la base de datos.</p> <p>Fracaso El Sistema descarta los cambios sin afectar la información anterior</p>

FLUJO NORMAL

- 1.- El administrador selecciona la opción “Editar” en el platillo seleccionado.
- 2.- El sistema muestra un formulario con la información del platillo y un campo para subir imagen.
- 3.- El administrador modifica los campos deseados.
- 4.- El administrador confirma la edición presionando “Confirmar”.
- 5.- El Sistema valida la información y muestra el mensaje “Modificado con éxito” y regresa a la pantalla principal.

Extensiones (Flujos Alternativos)

3a. El Administrador selecciona “Agregar imagen”:

1. El Sistema abre el explorador de archivos / galeria.
2. El Administrador selecciona la imagen
- 2a. El formato de la imagen no es válido:
 1. El Sistema muestra un mensaje “Formato de imagen no permitido”
 - Regresa al paso 3 del flujo normal
3. El Sistema guarda la imagen seleccionada.

**a. En cualquier momento del flujo normal, el Administrador presiona “Cancelar”:*

1. El Sistema pregunta si está seguro de cancelar
 - 1a.- El Administrador presiona “Aceptar”.
 - El Sistema descarta los cambios y regresa a la pantalla principal. Termina el caso de uso
 - 1b.- El Administrador presiona “Cancelar”.
 - Regresa al paso 2 del flujo normal

Flujos de Excepción

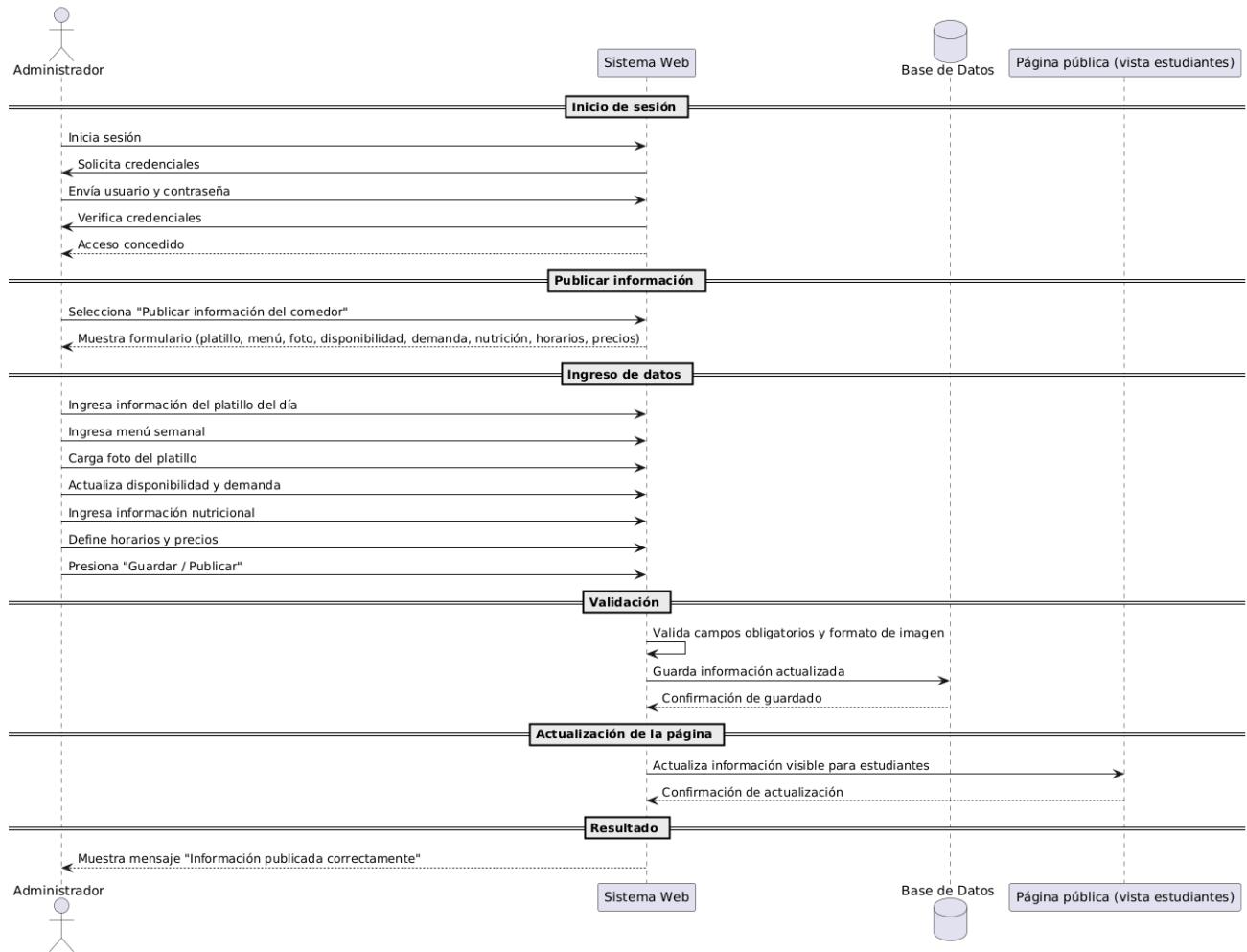
5b. Error al guardar la información

1. El sistema notifica “Error al guardar los datos. Intente nuevamente más tarde”.

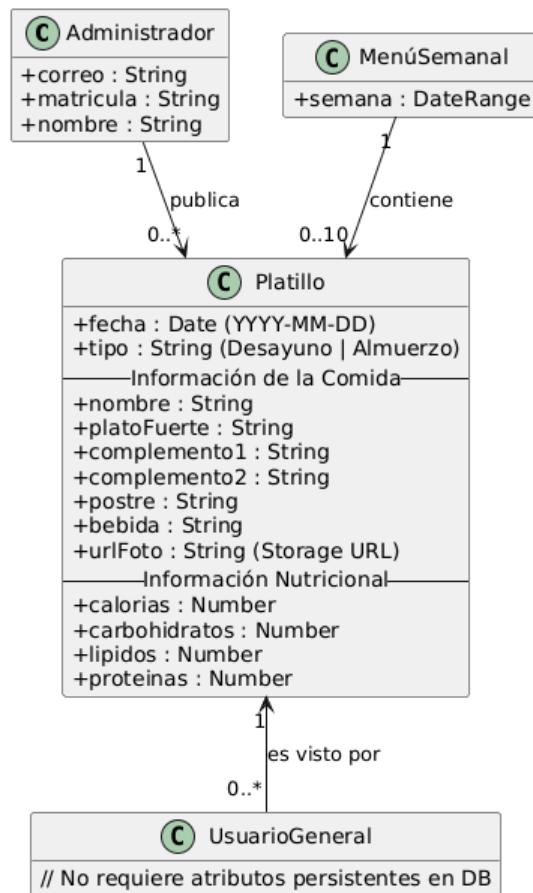
**a. En cualquier momento el Administrador sale inesperadamente del sistema o la conexión falla*

1. El Sistema descarta los cambios y devuelve a la pantalla principal cuando se restablezca la conexión

Caso de uso: Publicar información del comedor universitario



Modelo de dominio actual - Iteracion 1



Fase III: Construction

Las herramientas utilizadas para el desarrollo del código

Framework: [Next.js](#)

Librerías principales utilizadas en [Next.js](#)

- useEffect
- useState
- Mantine
- heroui
- Librerías para mantener conexión con la base de datos
- yup (importante para realizar validaciones)

Base de datos:

- Firebase auth implementado para la autenticación de los administradores encargados de alimentar de información la aplicación.
- Firebase Real Time Database Implementado para el almacenamiento de los platillos categorizados por el día de la semana y si es de tipo desayuno o comida. Es una base de datos no relacional, por lo que almacena la información en formato JSON con la estructura de un árbol
- Fiebase Storage para el almacenamiento de las fotografías de los platillos independientemente de lo pesado que esté la imagen.

Despliegue de la aplicación

Inicio rápido

- Crea una nueva aplicación Next.js llamada ComedorUAZ
- cd my-appe iniciar el servidor de desarrollo.
- Visita <http://localhost:3000>.

```
npm install      - + X
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y
mejoras. https://aka.ms/PSWindows

PS C:\Users\madri\Downloads\ejemplo> npx create-next-app@latest comedoruaz
✓ Would you like to use the recommended Next.js defaults? » No, customize settings
✓ Would you like to use TypeScript? ... No / Yes
✓ Which linter would you like to use? » ESLint
✓ Would you like to use React Compiler? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like your code inside a 'src/' directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the import alias ('@/*' by default)? ... No / Yes
✓ What import alias would you like configured? ... @/*
Creating a new Next.js app in C:\Users\madri\Downloads\ejemplo\comedoruaz.

Using npm.

Initializing project with template: app-tw

Installing dependencies:
- next
- react
- react-dom

Installing devDependencies:
- @tailwindcss/postcss
- @types/node
- @types/react
- @types/react-dom
- babel-plugin-react-compiler
- eslint
- eslint-config-next
- tailwindcss
- typescript
```

```

added 358 packages, and audited 359 packages in 2m

141 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

Generating route types...
✓ Types generated successfully

Initialized a git repository.

Success! Created comedoruaz at C:\Users\madri\Downloads\ejemplo\comedoruaz

PS C:\Users\madri\Downloads\ejemplo> cd comedoruaz
PS C:\Users\madri\Downloads\ejemplo\comedoruaz> npm run dev

> comedoruaz@0.1.0 dev
> next dev

  ▲ Next.js 16.0.6 (Turbopack)
  - Local:          http://localhost:3000
  - Network:        http://192.168.56.1:3000

  ✓ Starting...
  ✓ Ready in 2.4s

```

Estructura de los archivos deployados del proyecto:

PS C:\Users\madri\OneDrive\Documentos\Proyectos\uazcomedor\ComedorUAZ> dir			
Mode	LastWriteTime	Length	Name
da---l	24/11/2025 12:48 a. m.		.next
da---l	24/11/2025 12:11 a. m.		lib
da---l	02/12/2025 06:17 p. m.		node_modules
da---l	26/11/2025 11:26 a. m.		pages
da---l	24/11/2025 12:11 a. m.		public
da---l	24/11/2025 12:11 a. m.		styles
-a---l	24/11/2025 12:11 a. m.	521	.gitignore
-a---l	24/11/2025 12:11 a. m.	483	eslint.config.mjs
-a---l	24/11/2025 12:11 a. m.	77	hero.ts
-a---l	24/11/2025 12:48 a. m.	259	next-env.d.ts
-a---l	02/12/2025 07:00 p. m.	427	next.config.ts
-a---	02/12/2025 06:17 p. m.	456539	package-lock.json
-a---l	26/11/2025 11:26 a. m.	1135	package.json
-a---l	24/11/2025 12:11 a. m.	414	postcss.config.mjs
-a---l	24/11/2025 12:11 a. m.	1940	README.md
-a---l	24/11/2025 12:11 a. m.	632	tsconfig.json

Librerías implementadas en el página de inicio

```
import React, { useEffect, useState } from "react";
import { Navbar, NavbarBrand, NavbarContent, Chip } from "@heroui/react";
import { Button, Card, Accordion, Loader } from "@mantine/core";
import { Carousel } from "@mantine/carousel";
import Image from "next/image";
import Head from "next/head";
import { useRouter } from "next/router";
import { db } from "@/lib.firebaseioClient";
import { ref, onValue } from "firebase/database";
import { FormData, MenuData } from "./admin";
import { IconPhoto } from "@tabler/icons-react";
```

Librerías implementadas en la página de administradores

```
pages > admin > index.tsx > ...
1  import React, { useEffect, useState } from "react";
2  import {
3    Navbar,
4    NavbarBrand,
5    NavbarContent,
6    Chip,
7    addToast,
8  } from "@heroui/react";
9  import { Button, Card, Accordion, Divider, ActionIcon, FileButton } from "@mantine/core";
10 import Image from "next/image";
11 import Head from "next/head";
12 import { useRouter } from "next/router";
13 import { Modal, TextInput, NumberInput, Select, Checkbox } from "@mantine/core";
14 import { Carousel } from "@mantine/carousel";
15 import { useDisclosure } from "@mantine/hooks";
16 import { useForm, Controller, useFieldArray } from "react-hook-form";
17 import { yupResolver } from "@hookform/resolvers/yup";
18 import * as yup from "yup";
19 import { IconEdit, IconPhoto, IconPlus } from "@tabler/icons-react";
20 import { useFirebaseForm } from "@/lib/useForm";
21 import { db, storage } from "@/lib.firebaseioClient";
22 import { ref, onValue } from "firebase/database";
23 import { uploadBytes, getDownloadURL, ref as refs } from "firebase/storage";
```

Implementación de la base de datos

Firebase Auth

Firebase Authentication es un servicio de Firebase que ofrece un sistema para autenticar a los usuarios de una aplicación de forma segura y sencilla, permitiendo que inicien sesión con diferentes métodos. Soporta una variedad de opciones, como correo electrónico y contraseña, autenticación por teléfono, y proveedores de identidad federada como Google, Facebook, Apple, X (Twitter) y GitHub. Además, proporciona SDKs, bibliotecas y una consola de administración para facilitar el desarrollo y la gestión de cuentas de usuario. Para este proyecto se utilizó el método de identificación por correo y contraseña.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID del usuario		
admin@gmail.com	✉️	23 nov 2025	2 dic 2025	b5NMQ1fZqvcrL5Xi9ja24ACs...		

Como se puede observar, Firebase genera un UID y tokens para que se pueda identificar alloguearse en cada página y comparar credenciales para validar los permisos.

Se configuraron algunas acciones para aumentar la seguridad de las políticas de la contraseña.

Modo de aplicación forzosa
Se trata del modo de aplicación forzosa de las políticas de contraseñas

Exigir aplicación
Los intentos de registro fallarán hasta que el usuario actualice su contraseña de acuerdo con tu política

Notificar aplicación
Permite a los usuarios registrarse con una contraseña que no cumple con los requisitos. Se devuelven los criterios que faltan para satisfacer la política

Opciones de los requisitos para las contraseñas

<input type="checkbox"/> Exigir caracteres en mayúscula	<input type="checkbox"/> Exigir caracteres en minúscula
<input type="checkbox"/> Exigir caracteres especiales	<input type="checkbox"/> Exigir caracteres numéricos
<input type="checkbox"/> Forzar la actualización durante el acceso	

Requisitos de longitud de contraseña
Longitud mínima de la contraseña

6

Longitud máxima de la contraseña

4096

Firebase Real Time Database

Firebase Realtime Database es una base de datos NoSQL alojada en la nube que almacena y sincroniza datos en tiempo real entre los usuarios de una aplicación. Almacena los datos en formato JSON, y cada vez que se modifica un dato, la base de datos notifica a todos los clientes conectados para que reciban la actualización automáticamente, sin que sea necesario recargar la aplicación.

Características principales

- **Sincronización en tiempo real:** Permite a múltiples usuarios colaborar y ver los cambios al instante, como en una aplicación de chat o de edición colaborativa.
- **Multiplataforma:** Se integra con los SDK de Firebase para iOS, Android, web y juegos, permitiendo a los desarrolladores crear aplicaciones multiplataforma con facilidad.
- **Sincronización automática sin conexión:** Utiliza una caché local en el dispositivo para guardar y publicar cambios cuando el dispositivo no tiene conexión. Cuando se restablece la conexión, los datos se sincronizan automáticamente.
- **Seguridad:** Ofrece un modelo de seguridad declarativo para controlar el acceso a los datos basándose en la identidad del usuario o en el patrón de los datos.
- **Base de datos NoSQL:** Los datos se estructuran en un formato de árbol JSON, no en tablas relacionales.
- **Escalabilidad:** Aunque ha sido superada en escalabilidad y consulta por Cloud Firestore (su sucesora), todavía es una opción viable para aplicaciones con menos datos (unos pocos GB) que cambian con frecuencia.

Desglose de un fragmento de la base de datos Real Time:

```
{  
  "platillos": {  
    "Lunes": {  
      "Desayuno": {  
        "bebida": "Agua de horchata",  
        "complemento1": "Frijoles ",  
        "complemento2": "Ensalada",  
        "demanda": "Alta",  
        "descripcion": "Plato de chilaquiles acompañado de ensalada y  
frijoles",  
        "disponible": true,  
        "nombre": "Chilaquiles ",  
        "nutricional": {  
          "calorias": 100,  
          "carbohidratos": 150,  
          "lipidos": 20,  
          "proteinas": 50  
        }  
      }  
    }  
  }  
}
```

```

        },
        "platoFuerte": "Chilaquiles verdes",
        "postre": "Yogurt",
        "precio": 25,
        "tipo": "Comida"
    },
},
"Martes": {
    "Comida": {
        "bebida": "Agua de horchata",
        "complemento1": "Ensalada",
        "complemento2": "Verduritas",
        "demanda": "Media",
        "descripcion": "Torta de adobada acompañada con salsa roja con ensalada, junto con una galleta y agua de horchata",
        "disponible": true,
        "imagen":
"https://firebasestorage.googleapis.com/v0/b/comedoruaz.firebaseiostorage.app/o/platillos%2FMartes%2FComida%2Ftorta_ahogada.jpg?alt=media&token=c924fb19-a31e-416b-9147-9c8aab8c5a8a",
        "nombre": "Torta ahogada",
        "nutricional": {
            "calorias": 200,
            "carbohidratos": 30,
            "lipidos": 101,
            "proteinas": 30
        },
        "platoFuerte": "Torta de adopada ahogada",
        "postre": "Galleta",
        "precio": 25,
        "tipo": "Comida"
    },
    "Desayuno": {
        "bebida": "Agua de limón",
        "complemento1": "Arroz",
        "complemento2": "Verduritas",
        "demanda": "Alta",
        "descripcion": "Tortas de Atún con guarnición de arroz en salsa verde",
        "disponible": true,
        "imagen":
"https://firebasestorage.googleapis.com/v0/b/comedoruaz.firebaseiostorage.app/o/platillos%2FMartes%2FComida%2Fchilaquiles-rojos.jpg?alt=media&token=7fa34e4f"
    }
}

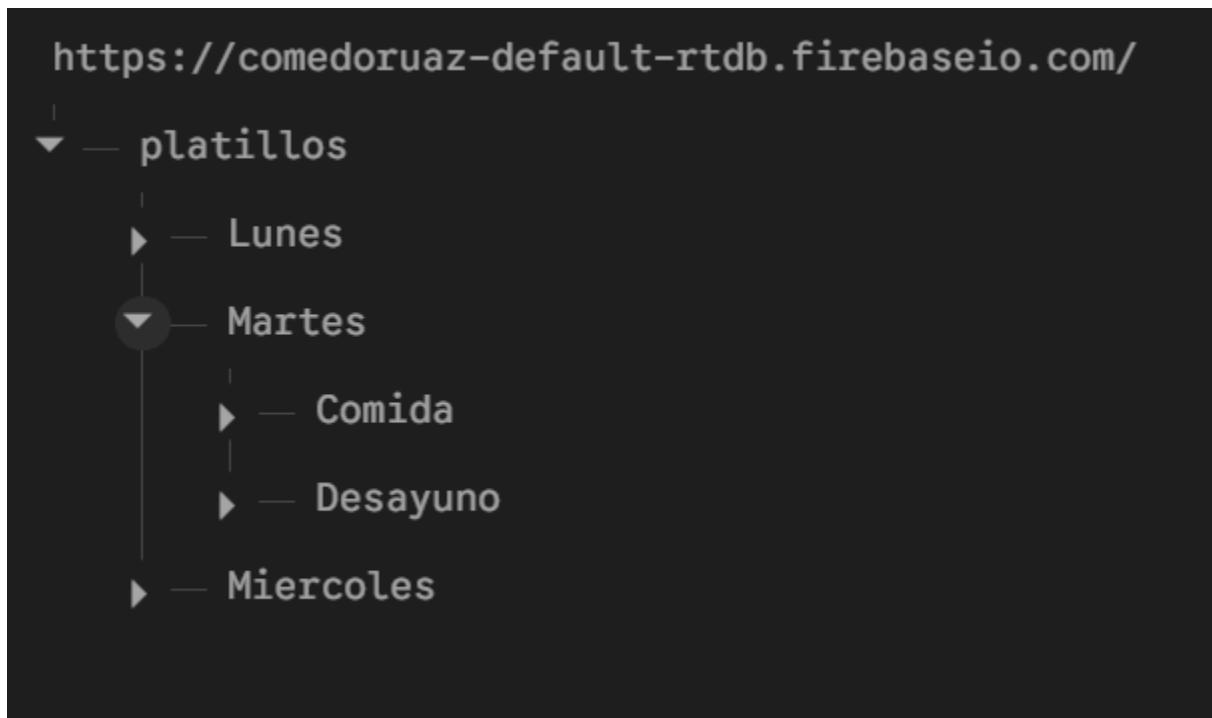
```

```

-203c-4ed3-ab37-3fcec1583d9c",
    "nombre": "Tortitas de Atún",
    "nutricional": {
        "calorias": 200,
        "carbohidratos": 30,
        "lipidos": 101,
        "proteinas": 30
    },
    "platoFuerte": "Tortitas de Atún con Salsa Verde",
    "postre": "Gelatina",
    "precio": 25,
    "tipo": "Comida"
}
}
}
}

```

Estructura en la base de datos a grandes rasgos



Firebase Storage

Firebase Storage es un servicio de almacenamiento en la nube que permite guardar y servir

archivos como imágenes, videos y audio. Utiliza Google Cloud Storage para almacenar los datos y ofrece la facilidad de subir y descargar archivos directamente desde las aplicaciones móviles y web usando sus SDKs. Sus principales ventajas incluyen la escalabilidad automática, integración con Firebase Authentication para seguridad, y la gestión de redes inestables.

Funcionamiento

Servicio de almacenamiento de objetos: Es una solución para almacenar y servir contenido generado por el usuario, como fotos y videos.

Almacenamiento en Google Cloud Storage: Los archivos se guardan en un bucket de Google Cloud Storage, lo que garantiza alta disponibilidad y redundancia global.

Acceso desde clientes: Los desarrolladores usan los SDK de Firebase para cargar y descargar archivos directamente desde el cliente (dispositivos móviles, navegadores web).

Gestión de red: Maneja automáticamente redes inestables, permitiendo que las operaciones se reanuden justo donde se quedaron si se interrumpe la conexión.

Beneficios clave

Seguridad: Permite configurar reglas de seguridad personalizadas para controlar el acceso y la modificación de archivos, protegiéndolos solo para usuarios autorizados.

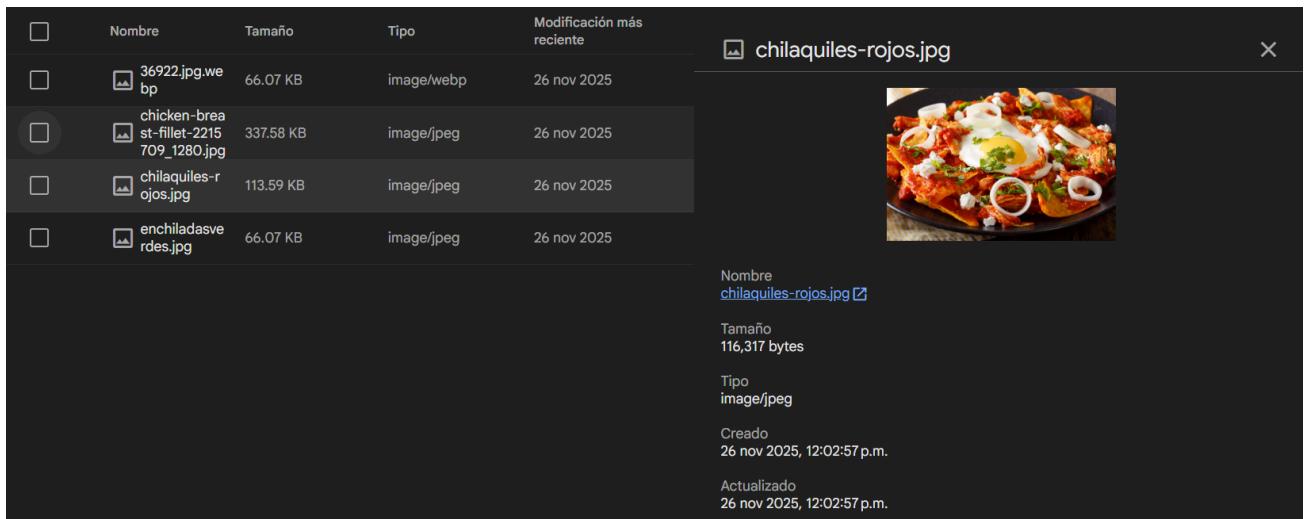
Escalabilidad automática: No es necesario preocuparse por la migración a un nuevo servicio a medida que el almacenamiento crece, ya que se escala automáticamente.

Integración: Se integra fácilmente con Firebase Authentication para controlar quién puede acceder a los archivos, y puede conectarse a Cloud Functions para realizar procesamiento en el servidor, como el filtrado de imágenes.

Facilidad de uso: Permite subir, descargar, eliminar y actualizar archivos de forma programática, y organiza los archivos en carpetas para una mejor administración.

Estructura de las imágenes almacenadas en Firebase Storage

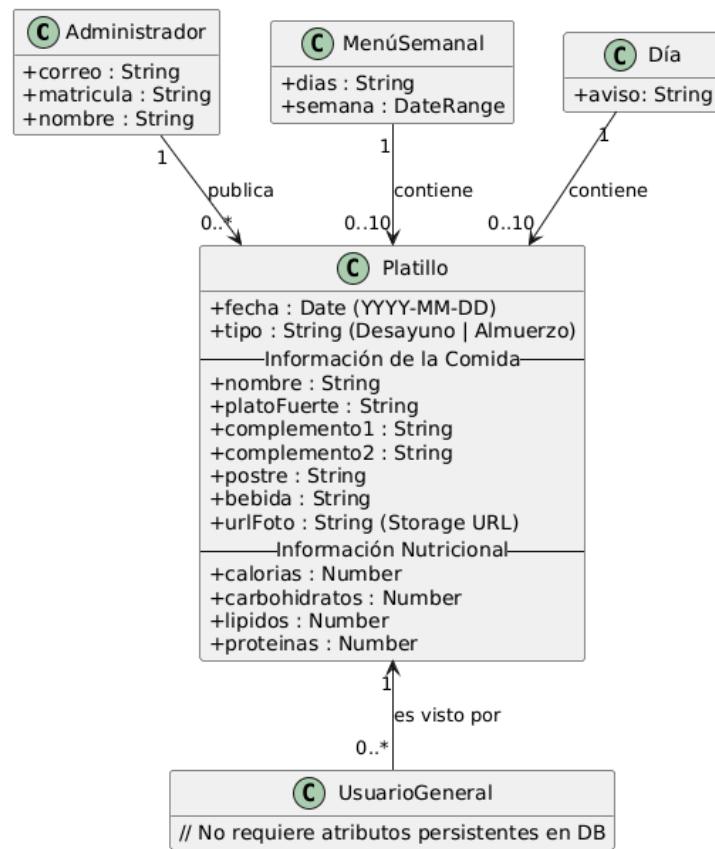
gs://comedoruz.firebaseio.storage.app > plátanos				
	Nombre	Tamaño	Tipo	Modificación más reciente
□	Jueves/	—	Carpeta	—
□	Martes/	—	Carpeta	—
□	Miercoles/	—	Carpeta	—



Próximos Pasos (Plan de Iteración 2)

- ACCIÓN (Ingeniero):** Generar el caso de uso de Registro Admin con una validación de personal autorizado
- ACCIÓN (Ingeniero):** Generar los casos de uso Ver Menu, Modificar Menú, y Agregar Menú, para que sea posible llevar control de menús que no necesariamente sean de la semana actual, incluyendo una funcionalidad de publicación automática de un menú ya guardado.
- ACCIÓN (Ingeniero):** Generar el caso de uso Agregar Aviso para que el administrador pueda proporcionar información relevante del servicio a los estudiantes como: días feriados, paros, cambios por disponibilidad de ingredientes.
- ACCIÓN (Ingeniero):** Ampliar el Modelo de Dominio generado en la iteración 1 para que permita el manejo de menús no necesariamente de la semana actual, así como agregar avisos puntuales sobre el servicio según sea requerido.

Propuesta de modelo de dominio para Próximas iteraciones



Fase IV: Transición

Visualización de la Aplicación

The screenshot displays the mobile application interface for the Comedor UAZ (UAZ Siglo XXI). The top navigation bar features the logo "SAUSU" and the text "Comedor UAZ" and "UAZ Siglo XXI".

The main content area shows a weekly meal plan:

- Martes:** Desayuno (Breakfast) - Tortitas de Atún (Tuna tortillas) - Available (Disponible). Description: Tortas de Atún con guarnición de arroz en salsa verde. Nutritional information: Calorías: 200, Proteínas: 30, Lípidos: 101, Carbohidratos: 30.
- Información nutrimental ^** (Nutritional information button) - Shows: Calorías: 200, Proteínas: 30, Lípidos: 101, Carbohidratos: 30.
- Menú semanal:** Miercoles (Wednesday) - Desayuno (Breakfast) - Chilaquiles rojos (Red chilaquiles). Description: Torta de adobada acompañada con salsa roja con ensalada, junto con una galleta y agua de horchata. Nutritional information: Calorías: 200, Proteínas: 30, Lípidos: 101, Carbohidratos: 30.
- Información nutrimental ^** (Nutritional information button) - Shows: Calorías: 200, Proteínas: 30, Lípidos: 101, Carbohidratos: 30.
- Comida:** Description: Enchiladas Verdes (Green enchiladas).

Alojamiento de la aplicación

Se alojó en una plataforma que no requiriera momentáneamente la compra de un dominio y ni un hosting.

<https://comedor-uaz.vercel.app/admin>



Ya se han realizado pruebas beta con estudiantes para la visualización del comedor y actualmente ya está disponible para todo el público que desee ingresar al link o escanear el QR.