



Instituto Tecnológico de Costa Rica

Escuela de Computación

Administración de proyectos

Proyecto del Curso

Profesor:

Jaime Solano Soto

Estudiantes:

Adrian Umaña Solano -2020426118

Andrés Valverde Barrios - 2020248159

Paola Melissa Alguera Castillo - 2019056061

Valeria Prado Rodriguez - 2020132773

Semestre II, 2022

Tabla de contenidos

Introducción:	3
Visión general	3
Definición del problema	4
Solución propuesta	5
Justificación	7
Descripción del documento	7
Descripción de la Escuela:	8
Nombre de la Escuela	8
Descripción general de la escuela	8
Organigrama	10
Contactos de la Escuela	10
Formulario de aprobación final del usuario	11
Ámbito del sistema:	13
Objetivos Generales	13
Objetivos Específicos	13
Criterios de Éxito	14
Alcances y suposiciones	15
Restricciones del proyecto	16
Funcionamiento	16
Base de datos:	17
Esquema de la base de datos	17
Descripción a Alto nivel	19
Reflexión:	20
Evaluación	20
Lecciones aprendidas	22
Errores y limitaciones conocidas	22
Instalación del sistema:	23
Manual de instalación	23
Apéndices:	23
Software Architecture Description (SAD)	23
Tablas de las bases de datos	25
Programación	26
Descripción de la capa de interfaz de usuario	26
Descripción de la capa de reglas de negocio (DLL)	28
Especificaciones del método	29

Introducción:

Visión general

Este documento es una extensión escrita de todo el trabajo del grupo de administración de proyectos del actual semestre II del 2022, en este archivo se ve reflejado el arduo trabajo del grupo y su aprendizaje a través de los meses.

En este escrito se encontrará la descripción del problema presentado y la solución programada propuesta por el equipo de trabajo tanto como web, como la solución programada móvil. La justificación del proyecto donde vendrán también los beneficios que trae la elaboración del proyecto.

Se darán datos de la escuela en la universidad a la que pertenece el proyecto y detalles exactos del funcionamiento de los algoritmos utilizados para el cumplimiento de los requisitos presentados.

Además se documentaran reflexiones, lecciones aprendidas , errores y limitaciones internas para consolidar el aprendizaje realizado por todos los miembros.

Definición del problema

En el año 2020 el mundo sufrió un cambio que nadie esperaba, una pandemia no antes vista por las generaciones actuales. Este acontecimiento obligó al planeta a una virtualización casi completa de servicios, interacciones, estudios, trabajos , entre muchos otros. Dentro de estos aspectos de la vida que se requirió cambiar tenemos el servicios de comidas express o solo de retiro en establecimientos de comida punto focal del proyecto planteado.

El ITCR brinda un servicio a la comunidad estudiantil y funcionarios internos de un comedor subsidiado para ayudarlos al acceso a todos los tiempos de comidas a un precio razonable. La vuelta a la presencialidad completa por parte de todo el estudiantado y funcionarios provoca una saturación de personas en horas específicas del día en el comedor estudiantil, brindando un ambiente idóneo para la propagación del virus COVID-19 y provocando también un desperdicio grande de tiempo en filas interminables para acceder a el tiempo de comido correspondiente.

Estos factores planteados anteriormente reflejan una problemática que aturde a las autoridades superiores del ITCR, se estima que si no se establecen medidas razonables para el problema del regreso a la normalidad relativa de la cotidianidad estudiantil, los campus universitarios serán fotos de contagios graves.

Se necesita llegar a una solución tecnológica al problema planteado para que la funcionalidad total del campus del Instituto Tecnológico de Costa Rica sea óptima y sin preocupaciones.

Solución propuesta

Dado que la cobertura de internet en nuestro país es tan amplia y estable se propone diseñar, desarrollar e implementar un aplicación móvil y una página web capaz de gestionar a todos los usuarios (estudiantes y funcionarios), gestionar los alimentos con sus respectivos tiempos de comida y ventas hechas desde alguno de estos dos medios desde cualquier lado del país.

Una aplicación donde puedas obtener el servicio del comedor sin necesidad de realizar largas filas y ocasionar un tráfico mayor de gente en la zonas mencionadas, una aplicación donde solo se compre la comida y se llegue a retirar al punto establecido por la institución.

Las funcionalidades generales que la propuesta plantea son las siguientes:

- La aplicación debe contener un apartado de inicio de sesión el cual debe de funcionar para 2 tipos de usuarios que se deben autenticar mediante un correo electrónico y una contraseña suministrados por el TEC, es decir, con el correo institucional: Administrador (puede ser un usuario predefinido en la base de datos) y el Cliente.
- El usuario administrador debe de gestionar alimentos agregar, eliminar o modificar en el sistema y los alimentos deben tener los atributos: ID, nombre del alimento, disponibilidad (si aún hay o no), tipo (Plato principal, Postre o adicional, bebida) y precio.
- El usuario administrador debe de gestionar clientes eliminar o modificar clientes en el sistema y ver su historial de compras.
- El usuario administrador gestiona los tiempos de comida donde podrá seleccionar qué alimentos estarán disponibles o no en los diferentes tiempos de comida del día (desayuno, almuerzo, media tarde, cena).

Cada alimento deberá ser asignado a uno de estos tiempos y solo se podrá comprar y reclamar en el mismo día, por tanto, no es válido para otros días.

- Y por último pero no menos importante el administrador podrá ver listado todos los pedidos realizados hasta el momento, se podrá visualizar el número de orden de compra, la fecha en que fue realizada, los detalles del usuario que efectuó la transacción (carné, cédula, nombre completo y correo electrónico) y los alimentos del pedido.
- El cliente podrán registrarse e iniciar sesión en la aplicación. En el proceso de registro se necesitarán los siguientes atributos: número de cédula, número de carné, nombre, apellido1 , apellido2, edad, fecha de nacimiento , correo electrónico y contraseña institucional. Al finalizar el registro se le enviará un correo al usuario con los datos de ingreso.
- El cliente puede comprar alimentos disponibles según el tiempo de comida y por su tipo además de agregar la cantidad que desean comprar y posteriormente se agregarán al carrito de compras.
- En el carrito de compras del cliente se podrá visualizar todos los productos añadidos (el precio de cada ítem, un precio total del plato principal sin adicionales o bebida, finalmente el total de la sumatoria de todos ellos incluyendo los adicionales y bebida) e incluso eliminar aquellos ítems que desee. Además de que se generará un pdf con los detalles de la compra al realizar el pago junto a un código qr para realizar el retiro del pedido.

Justificación

Durante un largo periodo de tiempo el Tecnológico de Costa Rica ha presentado una problemática con respecto a su manera de suministrar alimentos a la población estudiantil y funcionarios del lugar. Las largas filas y aglomeraciones constantes en el comedor del TEC representan una pérdida de tiempo y exposición a la problemática mundial de salud, esta aplicación será desarrollada para presentar una alternativa eficiente y concreta a estos pesares que el alumnado y sus funcionarios sufren diariamente.

TEC's Dining Room ofrece una solución eficaz a esto, permitiendo al cliente realizar todo el proceso por medio de su celular o buscador web, lo único que este necesita hacer una vez la orden es confirmada es dirigirse a la instalación para retirar su pedido.

Al día de hoy se tiene varias alternativas similares a la solución presente. Programas como Uber Eats, Didi y Rappi ofrecen maneras eficaces y cómodas de comprar comida sin necesidad de movernos de donde estemos, sin embargo, el comedor institucional no aparece como una de las opciones en estas aplicaciones y el resto de locales disponibles se encuentran a distancias mayores del TEC, lo cuál hace que el tiempo de espera sea mucho mayor y que las cuotas de entrega y servicio sean más elevadas, esto sin mencionar además el alto precio de los alimentos ofrecidos.

Descripción del documento

Este documento está dividido en 7 partes generales que se dividen en subtemas relevantes para el proyecto:

Introducción con sus partes:

- Visión general
- Definición del problema

- Solución propuesta
- Justificación
- Descripción del documento

Descripción de la escuela:

- Nombre de la Escuela
- Descripción general de la escuela
- Organigrama
- Contactos de la Escuela
- Formulario de aprobación final del usuario

Ámbito del sistema

- Objetivos
 - General
 - Específicos
- Criterios de Éxito
- Alcances y suposiciones
- Restricciones del proyecto
- Funcionamiento

Base de datos:

- Esquema de la base de datos
- Descripción a alto nivel
- Restricciones de integridad referencial

Reflexión

- Evaluación
- Lecciones aprendidas
- Errores y limitaciones conocidas

Instalación del sistema:

- Manual de instalación

Apéndices:

- Software Architecture Description (SAD): Solamente se requiere diseño de la arquitectura de software o el diagrama de arquitectura.
- Tablas de las bases de datos
- Programación
 - Descripción de la capa de interfaz de usuario
 - Descripción de la capa de reglas de negocio (DLL)
 - Tipos de datos abstractos
- Especificaciones del método

Descripción de la Escuela:

Nombre de la Escuela

Escuela de Ingeniería en Computación Instituto Tecnológico de Costa Rica

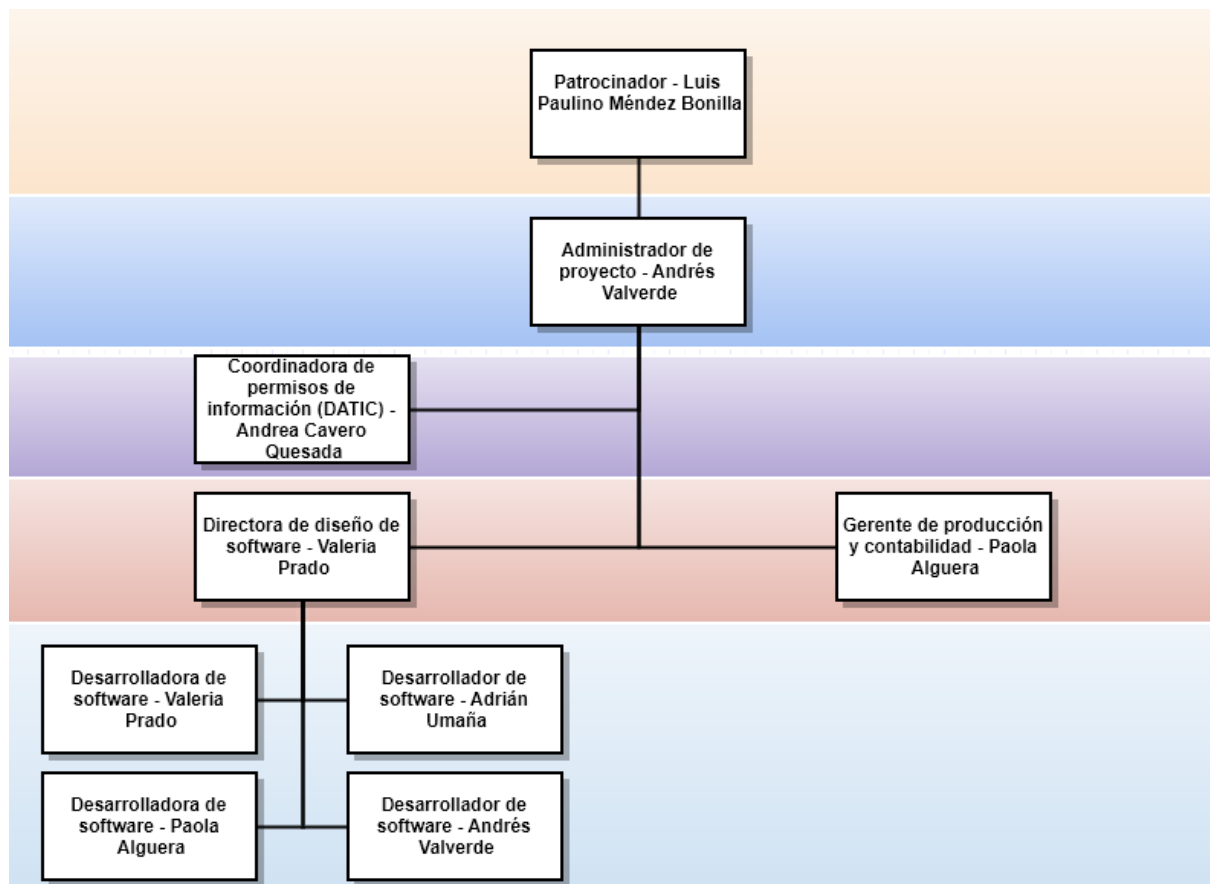
Descripción general de la escuela

La Escuela de Ingeniería en Computación ha creado un sólido prestigio como centro de excelencia en la enseñanza y en la investigación aplicada. Ofrece opciones académicas en los Campus Cartago, San Carlos, San José y en los Centros Académicos de Alajuela y Limón.

También cuenta con el Centro de Investigaciones en Computación (CIC), una entidad cuyo propósito es contribuir a solucionar problemas en la región de Centroamérica y el Caribe, mediante la realización de actividades orientadas a la generación, adaptación, incorporación y difusión de conocimientos informáticos.

La Escuela de Computación, también es partícipe en los programas de Administración de Tecnologías de Información (ATI) y de Ingeniería en Computadores (CE).

Organigrama



Contactos de la Escuela

Nombre	Correo electrónico	Posición
Andrés Valverde	andres190601@estudiantec.cr	Miembro del equipo
Valeria Prado	valeria44a@estudiantec.cr	Miembro del equipo
Adrián Umaña	adrian.s@estudiantec.cr	Miembro del equipo
Paola Alguera	paola.alguera@estudiantec.cr	Miembro del equipo

Jaime Solano Soto	jaimess@itcr.ac.cr	Profesor
Erika Marin Schumann	eshuman@itcr.ac.cr	Directora de carrera

Formulario de aprobación final del usuario

A continuación se establece cuales entregables de proyecto han sido aceptados:

Entregable	Aceptación (Si o No)	Observaciones
Aplicación Móvil funcional	Si	
Aplicación web funcional	Si	
Documentación	Si	

Para cada entregable aceptado, se da por entendido que:

El entregable ha cumplido los criterios de aceptación establecidos en la documentación de requerimientos y definición de alcance.

- Se ha verificado que los entregables cumplen los requerimientos.
- Se ha validado el cumplimiento de los requerimientos funcionales y de calidad definidos.
- Se ha realizado la transferencia de conocimientos y control al área operativa.
- Se ha concluido el entrenamiento que se definió necesario.
- Se ha entregado la documentación al área operativa.

Aprobación

Patrocinador	Fecha	Firma
Luis Paulino Méndez Bonilla	14-10-2022	<i>Luis Paulino Méndez Bonilla</i>

Ámbito del sistema:

Objetivos Generales

Desarrollar una aplicación web y móvil que permita hacer uso de los servicios del comedor institucional del tecnológico de Costa Rica antes del 2 de noviembre del 2022 y con un presupuesto de 10 millones de colones.

Objetivos Específicos

1. Gestionar a los usuarios de la aplicación y sus compras
2. Organizar las comidas ofrecidas en los distintos tiempos disponibles.
3. Administrar los pedidos que ya han sido completados.
4. Registrar a los usuarios de la aplicación por medio del correo institucional.
5. Sintetizar el proceso de compra y adquisición de alimentos de forma remota.

Criterios de Éxito

Número de objetivo específico	Criterio de éxito	Porcentaje logrado
1	Visualización de los cambios realizados a sus usuarios y compras en tiempo real.	100%
2	Creación del menú completo de un día (desayuno, almuerzo, media tarde y cena) en 5 minutos o menos.	100%
3	30000 pedidos exitosos un mes después del despliegue de la aplicación.	100%
4	5000 estudiantes registrados dos meses después de que la aplicación sea desplegada.	100%
5	Proceso y confirmación de la compra de una comida completa y generación del comprobante en 40 segundos o menos.	100%

Alcances y suposiciones

El alcance del proyecto se puede describir con los siguientes objetivos; que deben cumplirse con sus funcionalidades en el tiempo establecido para ser considerados exitosos, además todos los cambios deben ser reflejados a tiempo real para todos los usuarios:

- Un inicio de sesión funcional para los usuarios y sus distintos roles, validando que la información institucional sea correcta para cada usuario registrado según la base de datos existente.
- La gestión de Alimentos debe poder realizarse por un usuario administrador, con la posibilidad de agregar, eliminar o modificar alimentos, así como su precio, existencia y otras características.
- Los clientes pueden ser consultados, eliminados o modificados por un administrador.
- Se debe guardar exitosamente el historial de pedidos de cada cliente suponiendo que las compras hayan sido efectuadas correctamente, sin posibilidad de devolución.
- Lograr una gestión completa de los tiempos de comida, donde el usuario administrador puede asignar alimentos y estos son mostrados según disponibilidad para la compra.
- Mantener un historial de los pedidos realizados, se podrá visualizar el número de orden de compra, la fecha en que fue realizada, los detalles del usuario que efectuó y los alimentos del pedido.
- Se debe poder consultar los alimentos por medio de filtros y realizar la compra validando la disponibilidad.
- Debe existir un carrito de compra funcional donde se pueden visualizar los

productos añadidos y confirmar la compra.

- El proceso de compra debe generar una factura y un código QR con todos los datos de esta; se debe enviar al correo del cliente.

Restricciones del proyecto

El número de personas en el equipo de desarrollo de software es bastante limitado, pueden presentarse eventualidades personales en los miembros del equipo que influyan considerablemente en el desarrollo y tiempo estimado de la aplicación.

El tiempo de entrega puede resultar en un calendario de trabajo bastante ajustado, por lo que la realización de pruebas puede verse afectada por falta de tiempo.

El funcionamiento de la aplicación depende estrictamente de la base de datos existente en la institución, un fallo en el sistema de esta base de datos afecta directamente todas las funciones de la aplicación.

Funcionamiento

El hardware mínimo necesario para el uso de la aplicación web en cualquier computadora y móvil android :

- Procesador: Intel Core 2 Duo E8400 o equivalente
- RAM: 4 GB
- VRAM: 1 GB
- GPU: no requerida

- 2 GB DE MEMORIA RAM (móvil)
- Procesador 1,5 GHz multinúcleo (32 bits o 64 bits) (móvil)
- GPU: Mali-T860 o superior (móvil)

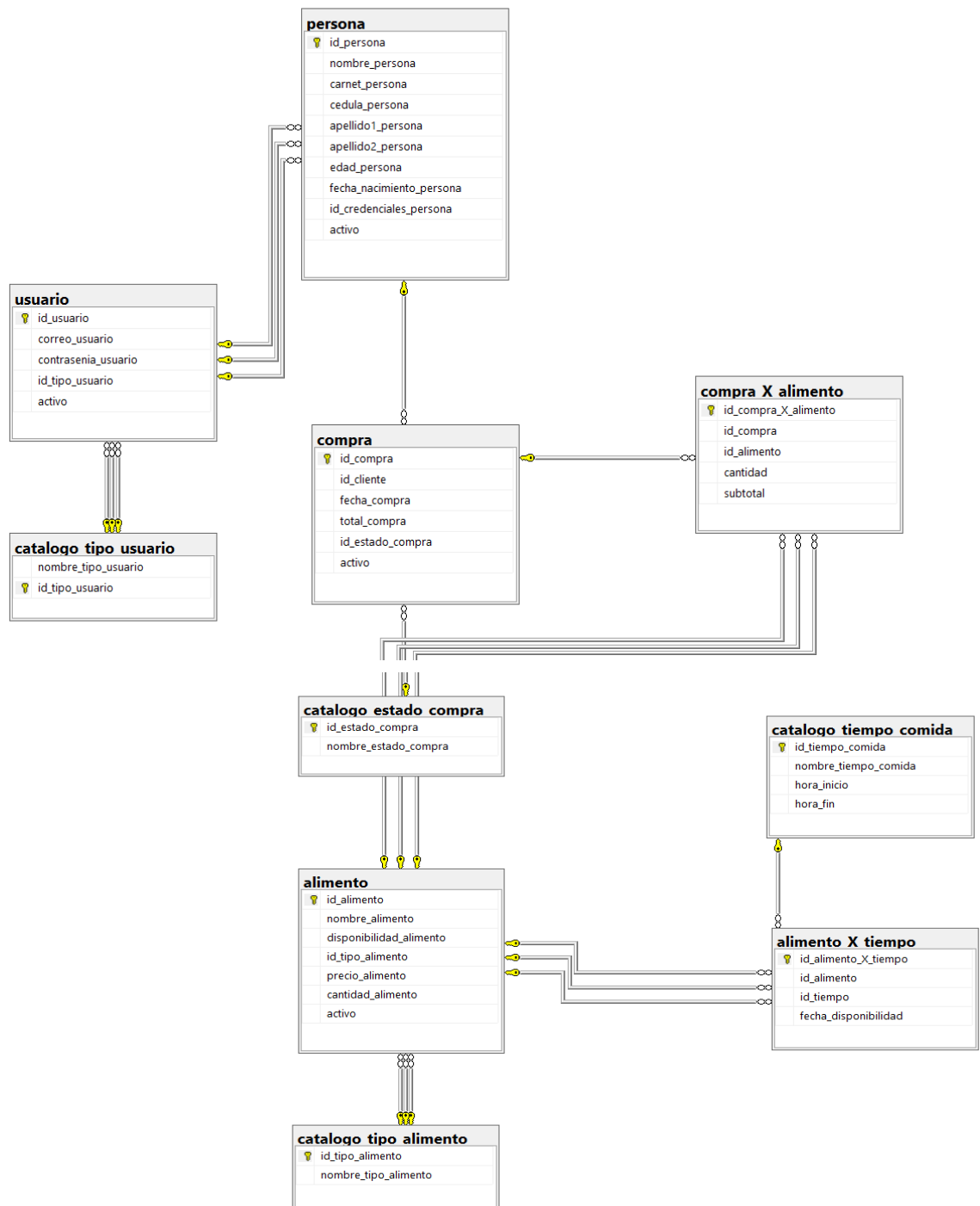
Se necesita un espacio de almacenamiento de 1,2 GB mínimo libre para el funcionamiento óptimo de la aplicación móvil.

El software mínimo necesario para el uso de la aplicación web en cualquier computadora y móvil android :

- SO: Windows 7/8/10/11 de 64 bits
- Mac OS Mojave 10.14.6 o superior
- Chrome 56.0.2924.76 o su equivalente para Mac.
- Android 5 o superior

Base de datos:

Esquema de la base de datos



Se usó Microsoft SQL Server como base de datos por que es la base de datos relacional en el que se posee un nivel más alto de experticia por parte del equipo de trabajo, además de poseer un IDE bastante sencillo y fácil de seguir, la estructura del proyecto y sus requerimientos son perfectos para el uso de una base

de datos relacional. El nivel de relación entre los datos no es complicado ni complejo otras características idóneas para usar SQL Server.

El esquema fue diseñado pensando en la relaciones entre la persona , compra y alimento que son las tres entidades más importantes por que de la conexión entre estas tres se desprende el funcionamiento de la aplicación.

Primeramente se planificó las tablas y los atributos que necesitaba cada tabla para cumplir con los requerimientos establecidos, después se procedió a establecer las relaciones y referencias entre ellas para construir los procedimientos almacenados y sus funciones correspondientes.

Descripción a Alto nivel

Tablas:

- Persona: Relación con usuario, ya que usuario es una Persona.
- Usuario: Usuario obtiene una referencia de catalogo_tipo_usuario necesario para saber que permisos tiene en la aplicación y sus datos son guardados junto con sus respectivos datos en Persona.
- catalogo_tipo_usuario :De esta tabla se obtiene el tipo de usuario que es (Cliente o Administrador) que es referenciado en usuario.
- Compra: tiene atributos que pertenecen a otras tablas como catalogo_estado_compra, compra_X_alimento y Persona , compra necesita saber a quien le pertenece de las personas registradas en la base de datos.
- Catalogo_estado_compra: compra referencia esta tabla.
- Compra_X_alimento: compra referencia esta tabla
- Alimento: Esta tabla es referenciada por compra_X_Alimento ya que es necesario saber cual y cuanto de un alimento es comprado, datos guardados en esta tabla.

- Catalogo_tipo_alimento: alimento referencia esta tabla.
- Catalogo_tiempo_comida: alimento_X_tiempo referencia esta tabla
- Alimento_X_tiempo: alimento referencia esta tabla al igual que catalogo_tiempo_comida, esta tabla necesita saber en que tiempo y que alimento está disponible.

Reflexión:

Evaluación

Actividad	Estado de la actividad del proyecto
Delimitar el alcance del programa.	Logrado exitosamente, el proyecto abarca lo justo para cumplir con todos los requerimientos
Determinar stakeholders.	Logrado exitosamente, se logró recolectar toda la información de los stakeholders y definir sus roles.
Realizar el análisis de los requerimientos.	Logrado exitosamente, a partir del análisis de requerimientos se logró establecer todas las funcionalidades en el programa.
Realizar análisis del presupuesto.	Se cumplió con el presupuesto establecido y los parámetros establecidos en gastos.
Realizar la definición del funcionamiento.	Se establecieron y delimitaron exitosamente todos los funcionamientos que tiene la aplicación que llevar a cabo.

Diseñar los componentes de la aplicación.	Diseño de los prototipos y componentes exitosos , apegado a las funcionalidades correspondientes.
Diseñar la base de datos.	Diseño de la base de datos exitoso , apegado a las funcionalidades correspondientes.
Implementar prototipo de la aplicación.	Implementación exitosa de prototipo, siguiendo todo lo previamente establecido por el equipo.
Crear la base de datos.	Implementación exitosa de la base de datos, siguiendo todo lo previamente establecido por el equipo.
Programar el backend de la aplicación móvil.	No hubieron contratiempos mayores a la hora de programar el backend de la aplicación móvil, por lo que la implementación fue exitosa.
Realizar la interfaz gráfica de la aplicación móvil.	No hubieron contratiempos mayores a la hora de programar la interfaz gráfica de la aplicación móvil, por lo que la implementación fue exitosa.
Programar el backend de la aplicación web.	No hubieron contratiempos mayores a la hora de programar el backend de la aplicación web, por lo que la implementación fue exitosa.
Realizar la interfaz gráfica de la aplicación web.	No hubieron contratiempos mayores a la hora de programar la interfaz gráfica de la aplicación web, por lo que la implementación fue exitosa.

Hosting para la aplicación web	Fallo a la hora de contar con hosting la aplicación web, se encuentra en servidores locales pero en correcto funcionamiento.
Solucionar los problemas o errores encontrados.	Depuración al 90%, no todos los casos de fallo fueron probados.
Escribir manuales de usuario.	No aplica

Lecciones aprendidas

- Antes de intentar resolver un problema, es bueno investigar si existen implementaciones que ya brindan la solución a nuestro problema. También, suele haber librerías que brindan herramientas para programar en adición a las funcionalidades básicas que provee el lenguaje en que se trabaja.
- El trabajo en equipo y la comunicación constante son factores clave para construir el proyecto de manera en que sea sencillo integrar las partes desarrolladas por los diferentes integrantes.
- Es muy importante la distribución de tiempos correcta para la óptima realización del proyecto y que por más sencillo que se vea una tarea no es bueno dejar correr el tiempo pensando que se hace “fácil”.

Errores y limitaciones conocidas

El desarrollo y funcionamiento de la aplicación será en español y su uso es estrictamente para personas que pertenezcan a la institución. Las funcionalidades como el cambio de idioma, el registro o compras de personas ajenas a la institución, sistema de devoluciones y reclamos, asignación automática de credenciales y sistema de envíos a domicilio no están contempladas en el desarrollo de la versión inicial.

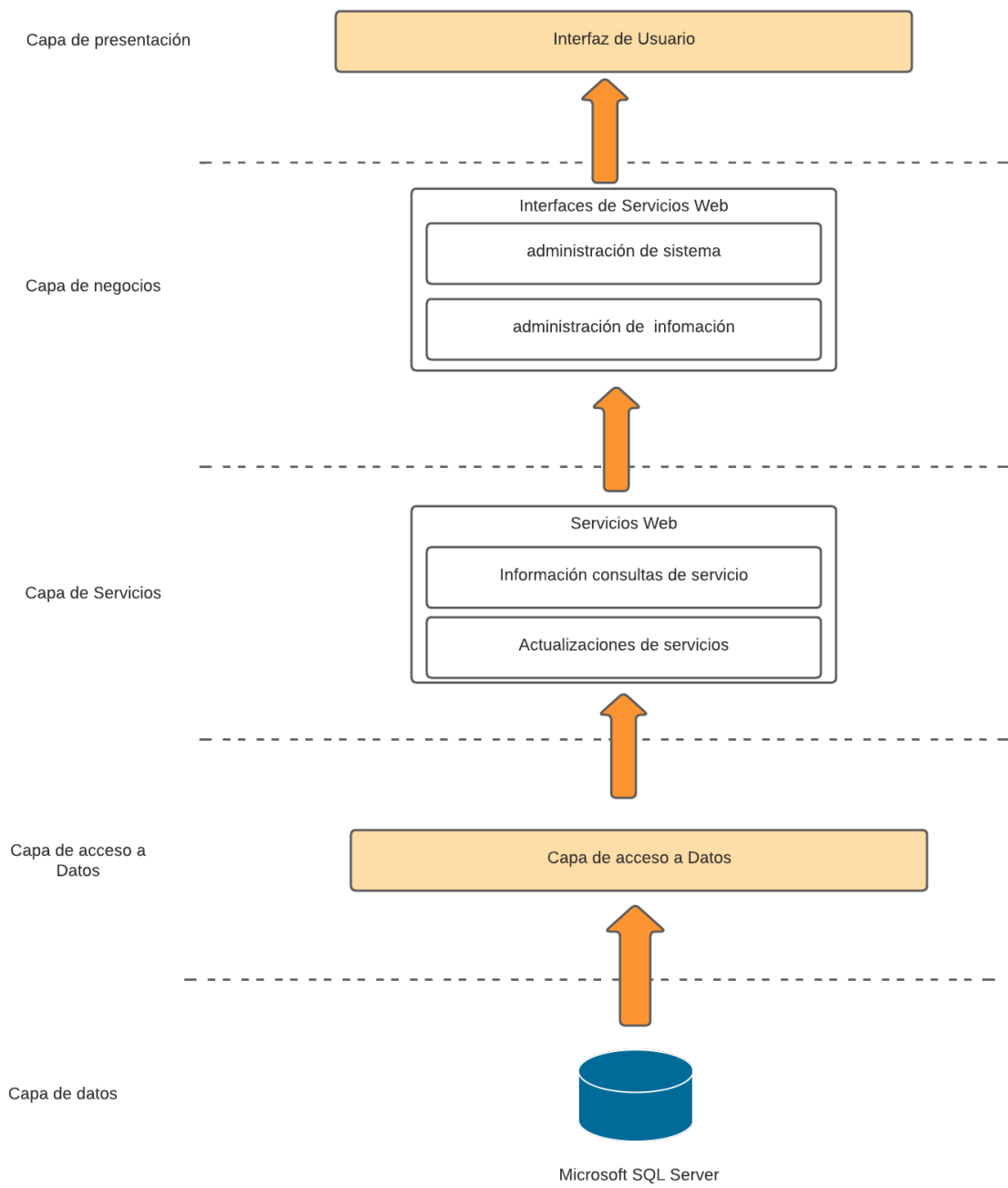
Instalación del sistema:

Manual de instalación

El manual de instalación se omite, ya que el enfoque de esta entrega no requiere uno, el entregable es una aplicación web por ende no es necesario.


Apéndices:

Software Architecture Description (SAD)




Tablas de las bases de datos


persona

	id_persona
	nombre_persona
	carnet_persona
	cedula_persona
	apellido1_persona
	apellido2_persona
	edad_persona
	fecha_nacimiento_persona
	id_credenciales_persona
	activo


usuario

	id_usuario
	correo_usuario
	contrasenia_usuario
	id_tipo_usuario
	activo


catalogo estado compra

	id_estado_compra
	nombre_estado_compra

catalogo tiempo comida

	id_tiempo_comida
	nombre_tiempo_comida
	hora_inicio
	hora_fin


catalogo tipo usuario

	nombre_tipo_usuario
	id_tipo_usuario

compra X alimento

	id_compra_X_alimento
	id_compra
	id_alimento
	cantidad
	subtotal

alimento

	id_alimento
	nombre_alimento
	disponibilidad_alimento
	id_tipo_alimento
	precio_alimento
	cantidad_alimento
	activo

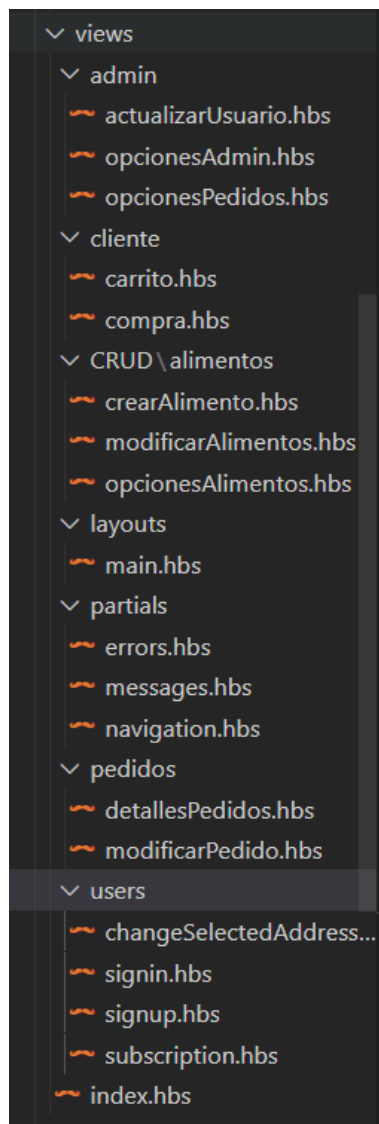
catalogo_tipo_alimento	
	id_tipo_alimento
	nombre_tipo_alimento

catalogo_tipo_usuario	
	nombre_tipo_usuario
	id_tipo_usuario

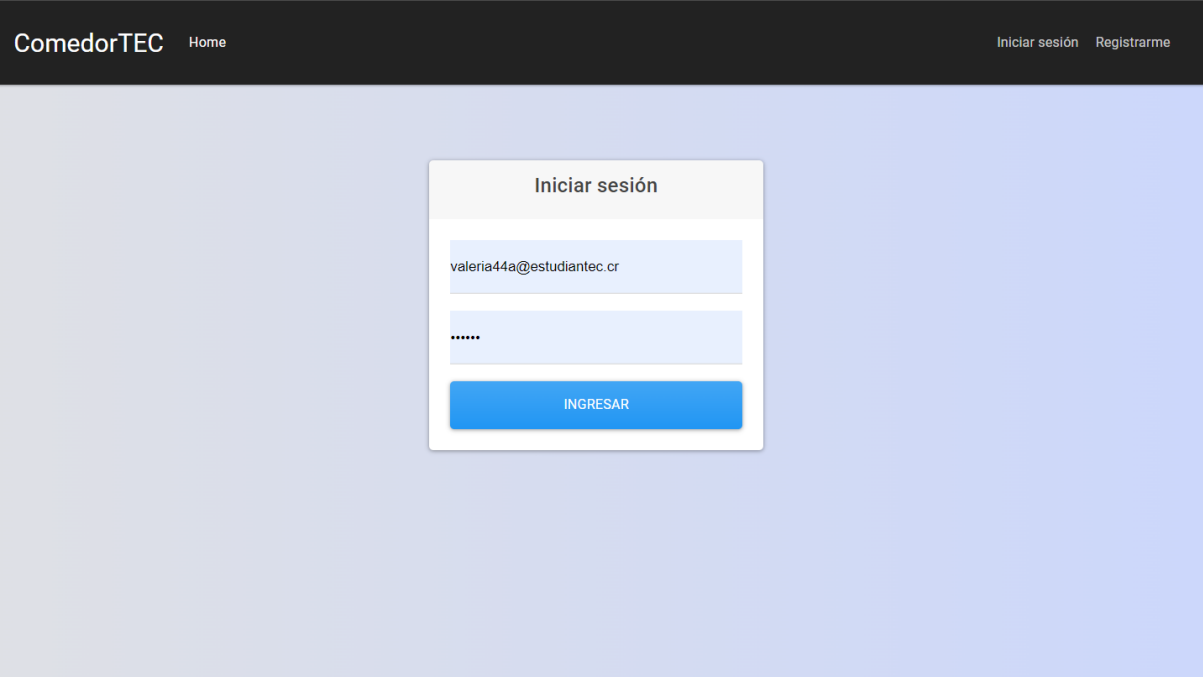
Programación

Descripción de la capa de interfaz de usuario

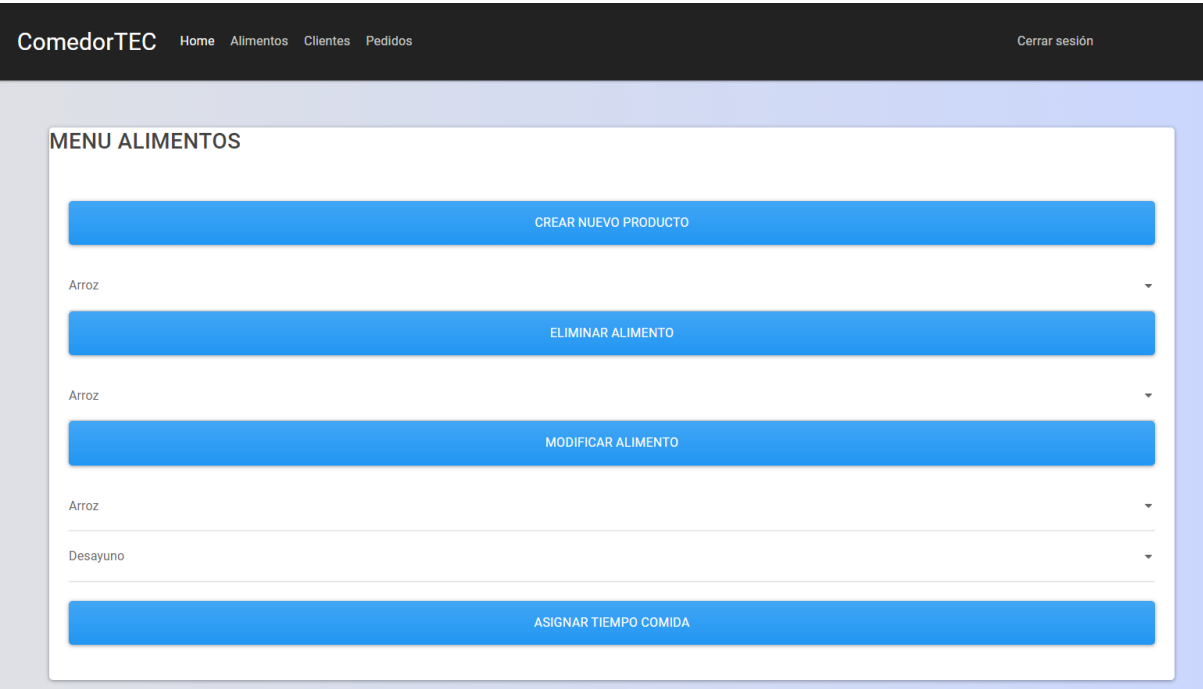
En esta capa se encuentran todas las pantallas utilizadas para el óptimo funcionamiento de la aplicación, como se puede apreciar en la siguiente imagen se ven todas las ventanas utilizadas en la aplicación.



El .hbs es una utilidad usada para programar esencialmente con html, cada pantalla está compuestas por un conjunto de componentes: inputs, botones, dropdowns, topbar y seleccionadores. En las pantallas se utilizan estos elementos a conveniencia de la funcionalidad que desempeña la vista. Se adjuntan imágenes de algunas vistas para demostrar la anteriormente mencionado.



Administrador:



Clientes:

ComedorTEC

HomeRealizar compraCarrito

Cerrar sesión

Comprar alimentos

Filtrar alimentos.

NO FILTRAR POR TIEMPO

NO FILTRAR POR TIPO

BUSCAR

INFORMACIÓN DEL ALIMENTO

ID	Nombre alimento	Disponible?	Tipo alimento	Tiempo	Precio en colones	Cantidad
7	Cas	true	Bebida	Almuerzo	300	1

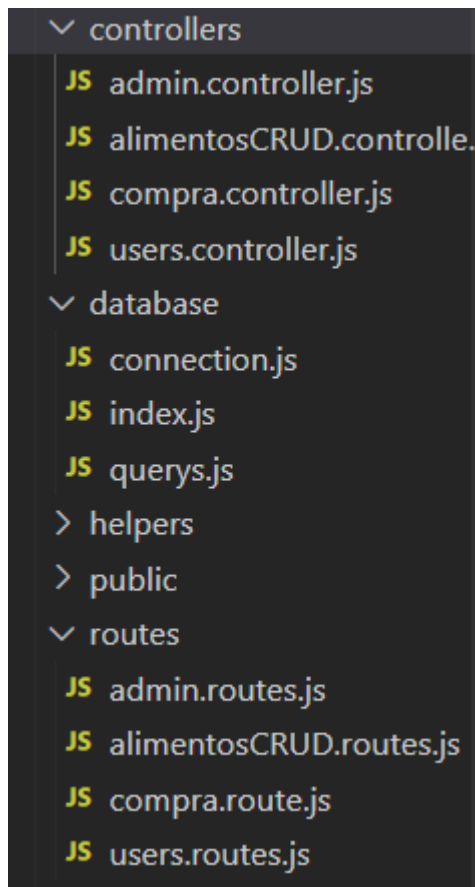
AGREGAR

Descripción de la capa de reglas de negocio (DLL)

Con respecto a la capa de reglas de negocios no se modificó gran cosa con respecto al entregable móvil. Se estableció un controlador para cada objeto primordial de la aplicación que a como su nombre lo dice posee los métodos necesarios para manipular la información que es necesaria.

También en esta capa se establece la conexión con la base de datos y algunas peticiones a la base importantes como:

- getAllProducts
- getAllUsers
- getProductByID
- addNewProduct
- getTotalProducts
- updateProductByID



Especificaciones del método

Se programaron métodos con una funcionalidad bastante parecida entre ellos para cargar las vistas y ejecutar procedimientos almacenados de la base de datos, a continuación se describen de forma general la funcionalidad de estos métodos.

Métodos para el administrador: from "../controllers/admin.controller";

- **const loadAdminMenu = async (req, res) =>** carga la vista general del menú para gestionar clientes del administrador con la información de los clientes de la base de datos, donde se puede seleccionar un cliente para eliminarlo o modificarlo.

- **const eliminarCliente = async (req, res) =>** recibe el carnet del cliente y ejecuta el procedimiento almacenado “borrarCliente” para eliminarlo de forma lógica.
- **const cargarModificarCliente = async (req, res) =>** recibe el carnet del cliente seleccionado y carga una vista con su información para modificarlo.
- **modificarCliente = async (req, res) =>** recibe los parámetros a modificar del cliente y ejecuta el procedimiento almacenado “modificarCliente”.
- **const loadMenuPedidos = async (req, res) =>** carga una vista con la información de todos los pedidos realizados, con la opción de filtrarlos, modificarlos o eliminarlos.
- **const loadMenuPedidosId = async (req, res) =>** en la misma vista anterior de pedidos carga los pedidos después del filtro por id.
- **const borrarPedido = async (req, res) =>** recibe un id del pedido y ejecuta el SP “eliminarPedido”.
- **const cargarDetallePedido = async (req, res) =>** recibe el ide de un pedido y carga su detalle o información (líneas de la factura).
- **cargarModificarPedido = async (req, res) =>** carga una vista con la información de un pedido seleccionado y los campos respectivos para modificar su información.
- **modificarPedido = async (req, res) =>** recibe los datos a modificar de un pedido y ejecuta el procedimiento almacenado “modificarPedido”.

Métodos del CRUD de alimentos: from "../controllers/alimentosCRUD.controller"

- loadAlimentosMenu = async (req, res) => carga una vista o menú con selects para la posibilidad de hacer un CRUD de alimentos o asignarlos a un tiempo de comida.
- agregarAlimento = async (req, res) => recibe como parámetros la información de un alimento y lo crea en la Base de Datos ejecutando el procedimiento “agregarAlimento”.
- eliminarAlimento = async (req, res) => recibe el id de un alimento seleccionado y lo elimina ejecutando el procedimiento almacenado “borrarAlimento”.
- modificarAlimento = async (req, res) => recibe los parámetros a modificar de un alimento y ejecuta el procedimiento almacenado “modificarAlimento”.
- asignarTiempoAlimento = async (req, res) => recibe el id del alimento y el id del tiempo seleccionado y ejecuta el SP “asignarTiempoComida” para asignar el alimento al tiempo seleccionado.

Metodos para el cliente: from "../controllers/compra.controller";

- getAlimentos = async (req, res) => carga una vista con la informacion de los alimentos disponibles en el tiempo de comida seleccionado.
- cargarBusqueda = async (req, res) => carga una vista con los alimentos disponibles en el tiempo de comida seleccionado filtrados por tipo.
- addProductToCart = (req, res) => añade un alimento al carrito de compras con su información en formato json.
- loadCarrito = (req, res) => carga una vista con los alimentos agregados previamente al carrito de compras.

- confirmarCompra = async (req, res) => ejecuta el procedimiento almacenado “generar compra” con el id del cliente y los alimentos de carrito de compras como parámetros, además genera un qr y pdf con la información de la factura y es enviado al correo del cliente. Vacía el carrito de compras.

Registrar usuario: from "../controllers/users.controller"

- const createNewUser = async (req, res) => recibe como parámetros la información de un cliente y ejecuta el procedimiento almacenado “CreateNewUser” para crear un nuevo usuario en la base de datos.