

UNIVERSIDAD DEL VALLE DE GUATEMALA

CAMPUS CENTRAL

PROGRAMACIÓN ORIENTADA A OBJETOS

Proyecto Entrega #2

Diego Gudiel - 24451

Damián Rodríguez - 24537

Juan Pablo Salguero - 24374

Pablo Toledo - 24355

Norman Aguirre - 24479

Sección: 20

Guatemala, 22 de Julio de 2024

Índice

Identificación de Requisitos Funcionales.....	4
Priorización de Requisitos Funcionales.....	5
Prioridad 1:.....	5
Prioridad 2:.....	5
Prioridad 3:.....	5
Identificación y Descripción de Clases Necesarias.....	6
Clase Pedido.....	6
Atributos:.....	6
Métodos:.....	6
Clase Restaurante.....	6
Atributos:.....	6
Métodos:.....	6
Clase InterfazUsuario.....	7
Atributos:.....	7
Métodos:.....	7
Clase ControladorPedido.....	7
Atributos:.....	7
Métodos:.....	7
Clase ControladorRestaurante.....	7
Atributos:.....	7
Métodos:.....	7
Diseño del Sistema.....	8
Diagrama de clases UML.....	8
Investigación Tecnologías Disponibles.....	9
Planificación y Gestión.....	10
Listado de requisitos a seguir.....	10
Gestión de Usuarios:.....	10
Gestión de Pedidos:.....	10
Selección de Restaurantes:.....	10
Visualización de Menús:.....	11
Notificaciones:.....	11
Especificación de Tareas.....	12
Creación y Gestión de Usuarios.....	12
Creación y Gestión de Pedidos.....	12
Implementación de Selección de Restaurantes.....	12
Implementación de Visualización de Menús.....	13
Implementación de Notificaciones.....	13
Calendario de planificación:.....	14
Formularios de cada integrante.....	16
Link repositorio en GitHub.....	17

Identificación de Requisitos Funcionales

Realizar pedidos en línea: Los usuarios podrán realizar pedidos de comida en línea desde cualquier restaurante del campus.

Ver estado del pedido: Los usuarios podrán visualizar el estado en tiempo real de sus pedidos.

Recibir notificaciones: Los usuarios recibirán notificaciones cuando el pedido esté listo para ser recogido.

Consultar menú actualizado: Los usuarios podrán consultar el menú actualizado de cada restaurante.

Evitar aglomeraciones: La aplicación indicará a los usuarios la cantidad de personas en los diferentes puntos de venta en tiempo real.

Priorización de Requisitos Funcionales

Prioridad 1:

Realizar pedidos en línea: Sin esta funcionalidad, la aplicación no cumpliría con su propósito principal.

Ver estado del pedido: Es crucial para que los usuarios sepan cuándo su pedido estará listo.

Recibir notificaciones: Es fundamental para avisar a los usuarios sobre el estado de su pedido.

Prioridad 2:

Consultar menú actualizado: Aunque importante, no es crítico para el funcionamiento inicial de la aplicación.

Prioridad 3:

Evitar aglomeraciones: Esta es una característica adicional que mejora la experiencia, pero no es esencial para la funcionalidad básica.

Identificación y Descripción de Clases Necesarias

Clase Pedido

Atributos:

idPedido: Identificador único del pedido.

estado: Estado actual del pedido (en proceso, listo, entregado).

tiempoEstimado: Tiempo estimado de preparación.

detalle: Detalles del pedido (productos seleccionados).

Métodos:

actualizarEstado(): Actualiza el estado del pedido.

calcularTiempoEstimado(): Calcula el tiempo estimado en base a la carga de trabajo.

Clase Restaurante

Atributos:

idRestaurante: Identificador único del restaurante.

menu: Menú disponible en el restaurante.

nombre: Nombre del restaurante.

Métodos:

actualizarMenu(): Actualiza el menú del restaurante.

consultarDisponibilidad(): Consulta la cantidad de pedidos en curso.

Clase InterfazUsuario

Atributos:

pantalla: Elementos visuales que se muestran al usuario.

Métodos:

mostrarMenu(): Muestra el menú del restaurante seleccionado.

mostrarEstadoPedido(): Muestra el estado actual del pedido.

Clase ControladorPedido

Atributos:

pedidoActual: El pedido que se está gestionando actualmente.

Métodos:

realizarPedido(): Inicia un nuevo pedido.

consultarEstado(): Consulta y actualiza el estado del pedido.

Clase ControladorRestaurante

Atributos:

restaurantesDisponibles: Lista de restaurantes disponibles.

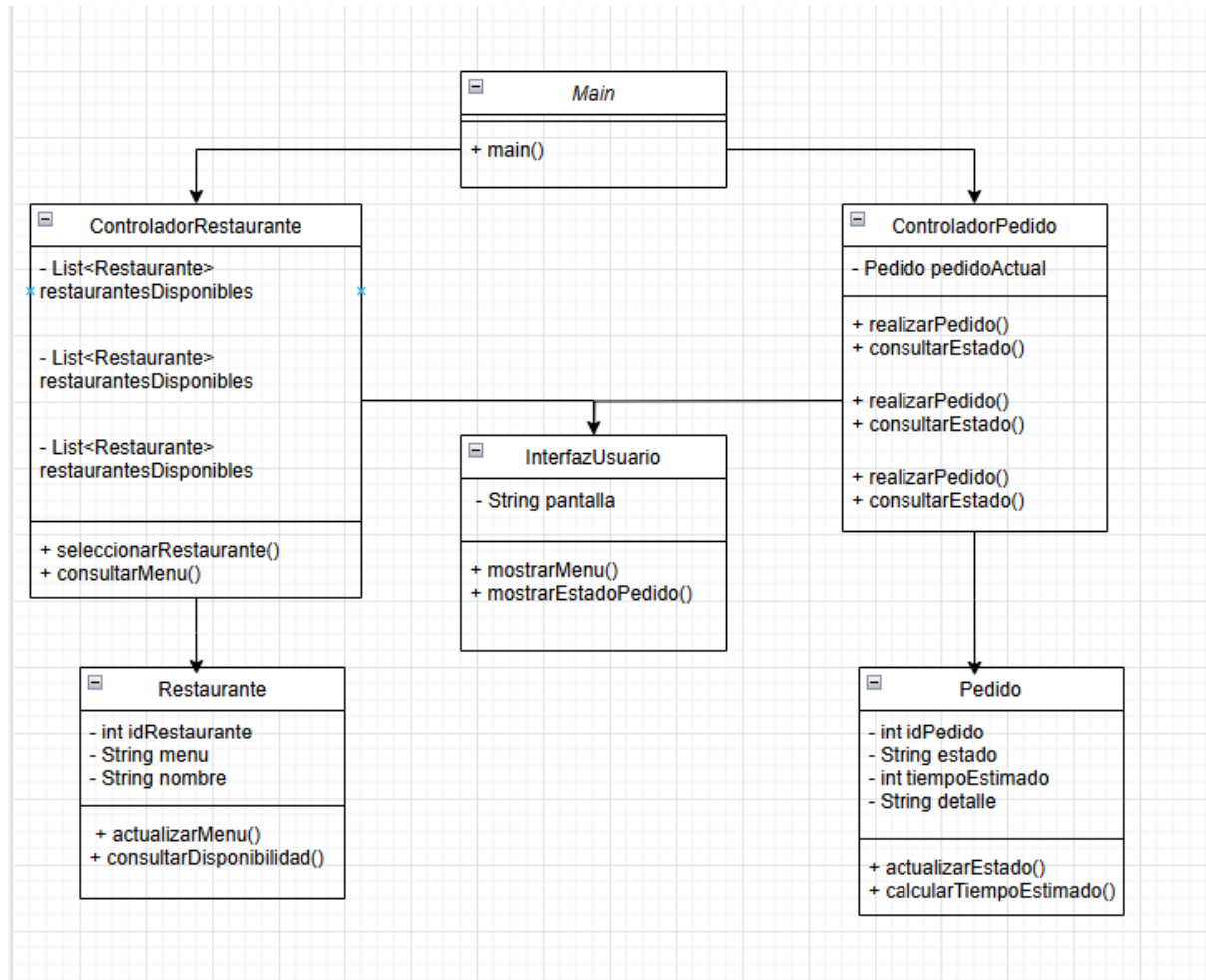
Métodos:

seleccionarRestaurante(): Selecciona el restaurante donde se realizará el pedido.

consultarMenu(): Consulta el menú de un restaurante.

Diseño del Sistema

Diagrama de clases UML



Investigación Tecnologías Disponibles

- Lenguaje de programación:
 - Java
- Framework:
 - Spring Boot
- Bases de datos:
 - MySQL
- Frontend:
 - React
- Gestión del proyecto:
 - Git

Se desarrollará una aplicación web con java como lenguaje principal. Dentro de las muchas opciones de frameworks para el proyecto, se estuvo decidiendo entre Spring Boot o Jakarta EE. Pero tras evaluar ambas opciones, se ha decidido que el más adecuado es Spring Boot, por su facilidad de uso y su cantidad de herramientas para el desarrollo web. Se integra fácilmente a las bases de datos y a las interfaces gráficas (VMware Tazu, 2024).

Para las bases de datos se utilizará MySQL. Se ha seleccionado MySQL por su eficiencia y uso de integración con Java. La interacción de MySQL y Java facilitará la interacción de la aplicación junto a su base de datos. En el frontend se va a utilizar React para la interfaz del usuario. Debido a la capacidad de crear interfaces dinámicas y de integrarse a Spring boot mediante APIs RESTful (Purohit, 2024).

Para gestionar el proyecto y facilitar el trabajo en equipo durante el desarrollo se utilizará Git. Esto nos permitirá tener un sistema de control de versiones y se podrá trabajar simultáneamente en diferentes partes del proyecto, con un trabajo más organizado y productivo.

Planificación y Gestión

Listado de requisitos a seguir

Gestión de Usuarios:

- Los estudiantes deben poder registrarse y acceder con credenciales seguras.
- ☐ Crear archivo .CSV con el que se guardaran los usuarios creados, así mismo poder llevar el control de sus gastos, sus formas de pago, etc.
- ☐ Crear una clase que deba poder realizar acciones con estos atributos de la clase usuario.
- ☐ Probar y/o modificar la persistencia de los datos del usuario.

Gestión de Pedidos:

- Los estudiantes deben poder crear y gestionar sus pedidos de comida a través de la aplicación.
- ☐ Creación de una clase para poder crear un objeto pedido, el cual tendrá toda la información del mismo, como el restaurante y productos seleccionados, el monto final, forma de pago, vuelto, etc.
- ☐ Crear una clase que esté constantemente verificando el estado del pedido, el cual mantendrá al usuario informado sobre el estado en tiempo real del pedido.
- ☐ Despliegue de la clase menú, para poder crear el pedido del usuario.

Selección de Restaurantes:

- La aplicación debe mostrar una lista de restaurantes disponibles en el campus.
- ☐ Crear clase Restaurantes, el cual tendrá la información de los mismos, como el menú de productos y precios, el tiempo promedio de espera para las entregas, recomendaciones, etc.
- ☐ Poder modificar los atributos de la clase, para que esta se pueda ir actualizando constantemente según el establecimiento lo considere necesario. Ya sea por cambios permanentes en el menú, como retiro o íntegro de platillos, así como cambios temporales, como ofertas de temporada.
- ☐ Verificación que la clase pueda crear y manejar los distintos tipos de restaurantes que hay dentro de la universidad, para no limitar el menú de las empresas.

Visualización de Menús:

- Los menús de los restaurantes deben estar disponibles para los estudiantes, mostrando opciones de comida, precios y disponibilidad.

- ☐ Crear un despliegue de opciones de los restaurantes, para que el usuario pueda elegir uno, y poder desplegar otro menú con lo que ofrece el establecimiento escogido.
- ☐ Deberá de poder manejar los cambios anteriormente mencionados, para así no crear un conflicto entre los menús.
- ☐ Darle prioridad a los platillos que están en oferta, ya que muchos estudiantes prefieren buscar el precio más económico.
- ☐ Verificar que los cambios en los objetos de la clase Restaurante no trunquen el funcionamiento de la clase visualización de menú.

Notificaciones:

- Los estudiantes deben recibir notificaciones sobre el estado de sus pedidos, incluyendo cuando el pedido esté listo para recoger.

- ☐ Debe crearse la clase Notificación_Usuario, el cual se encargará de notificar al usuario que su pedido está listo para recoger en el restaurante.
- ☐ Debe crearse la clase Notificación_Restaurante, el cual debe notificar al restaurante que tiene un pedido a realizar, con los datos del pedido, y que debe informar que en el momento que se encuentre listo para entregar (así poder notificar al usuario).

Especificación de Tareas

Creación y Gestión de Usuarios

Descripción de la tarea: Crear un archivo .CSV para almacenar la información de los usuarios, incluyendo datos como gastos y formas de pago. Desarrollar una clase que maneje los atributos del usuario, permitiendo realizar acciones como el registro y la autenticación. Implementar y probar la persistencia de datos de usuario, asegurando que la información se guarde y recupere correctamente.

Horas estimadas de desarrollo: 20 horas

Responsable de desarrollarla: Norman Aguirre

Fecha probable de terminación de la tarea: 3 de noviembre de 2024

Creación y Gestión de Pedidos

Descripción de la tarea: Desarrollar una clase para manejar los pedidos, incluyendo detalles como restaurante, productos seleccionados, monto final, forma de pago, etc. Crear una clase que verifique constantemente el estado del pedido y lo actualice en tiempo real. Implementar la clase que despliegue el menú para la creación del pedido por parte del usuario.

Horas estimadas de desarrollo: 23 horas

Responsable de desarrollar: Damián Rodríguez

Fecha probable de terminación de la tarea: 3 de noviembre de 2024

Implementación de Selección de Restaurantes

Descripción de la tarea: Crear la clase Restaurantes que contenga la información necesaria sobre cada establecimiento, como menú, tiempos de espera, recomendaciones, etc. Implementar funciones para modificar los atributos de la clase y permitir actualizaciones constantes según las necesidades del establecimiento. Verificar la capacidad de la clase para manejar múltiples tipos de restaurantes dentro de la universidad.

Horas estimadas de desarrollo: 19 horas

Responsable de desarrollarla: Pablo Toledo

Fecha probable de terminación de la tarea: 3 de noviembre de 2024

Implementación de Visualización de Menús

Descripción de la tarea: Desarrollar la funcionalidad que despliegue opciones de restaurantes para que el usuario pueda seleccionar uno y visualizar el menú correspondiente. Asegurar que la clase pueda manejar cambios en el menú, evitando conflictos. Dar prioridad a los platillos en oferta, verificando que los cambios en los objetos de la clase Restaurante no afecten la funcionalidad de la visualización del menú.

Horas estimadas de desarrollo: 21 horas

Responsable de desarrollarla: Juan Pablo Salguero

Fecha probable de terminación de la tarea: 3 de noviembre de 2024

Implementación de Notificaciones

Descripción de la tarea: Crear la clase Notificación_Usuario que notificará al usuario cuando su pedido esté listo para ser recogido. Desarrollar la clase Notificación_Restaurante que notificará al restaurante sobre los pedidos pendientes y cuándo están listos para la entrega.

Horas estimadas de desarrollo: 22 horas

Responsable de desarrollarla: Diego Samuel Gudiel

Fecha probable de terminación de la tarea: 3 de noviembre de 2024

Calendario de planificación:

Fecha de Terminación	Tarea	Parte/Función	Responsable
29/09/2024	Creación y Gestión de Usuarios	Prototipo del archivo `.CSV` con estructura básica	Norman Aguirre
06/10/2024	Creación y Gestión de Usuarios	Implementación básica de la clase `Usuario` con funciones de registro	Norman Aguirre
06/10/2024	Creación y Gestión de Pedidos	Prototipo de la clase `Pedido` con estructura básica	Damián Rodríguez
06/10/2024	Implementación de Selección de Restaurantes	Prototipo de la clase `Restaurantes` con datos estáticos iniciales	Pablo Toledo
06/10/2024	Implementación de Visualización de Menús	Prototipo de la funcionalidad de despliegue de menús	Juan Pablo Salguero
06/10/2024	Implementación de Notificaciones	Prototipo de la clase con `Notificación_Usuario` notificaciones básicas	Diego Samuel Gudiel
13/10/2024	Creación y Gestión de Usuarios	Prueba y modificación de la persistencia de datos	Norman Aguirre
13/10/2024	Creación y Gestión de Pedidos	Implementación de la verificación del estado del pedido en tiempo real	Damián Rodríguez
13/10/2024	Implementación de Selección de Restaurantes	Implementación de funciones para modificar y actualizar la clase `Restaurantes`	Pablo Toledo
13/10/2024	Implementación de Visualización de Menús	Implementación de manejo de cambios en los menús	Juan Pablo Salguero

13/10/2024	Implementación de Notificaciones	Implementación de la clase `Notificación_Restaurante`	Diego Samuel Gudiel
20/10/2024	Creación y Gestión de Pedidos	Finalización de la gestión completa de pedidos	Damián Rodríguez
20/10/2024	Implementación de Selección de Restaurantes	Verificación y finalización de la clase `Restaurantes`	Pablo Toledo
27/10/2024	Implementación de Visualización de Menús	Finalización de la visualización de menús con priorización de platillos en oferta	Juan Pablo Salguero
27/10/2024	Implementación de Notificaciones	Finalización de todas las notificaciones y pruebas de integración	Diego Samuel Gudiel
03/11/2024	Integración Final	Integración completa de todas las partes y pruebas finales	Todo el equipo

[illegible][illegible]



Link repositorio en GitHub

<https://github.com/Naguirrel/UVG-Eats>

Referencias

VMware Tanzu. (2024). Spring <https://spring.io>

Purohit R. (9 de agosto de 2024). A Step-by-Step Guide to Implementing React Spring Boot in Your Application. <https://www.dhiwise.com/post/a-step-by-step-guide-to-implementing-react-spring-boot>