

Universidad Autónoma de Yucatán

Facultad de Matemáticas

Construcción de software

5to semestre, grupo B

Integrantes:

Russel Bonilla Pech

Jaqueline Góngora Tun

Frida Pineda Alvarado

Víctor Rosado Koyoc

Luisa Villanueva Díaz

Profesor: Edwin León

Fecha de entrega: 15 de Noviembre de 2024

Plan de Trabajo

Descripción General:

El proyecto consiste en desarrollar un sistema de administración para una cafetería utilizando Java. Este sistema permitirá gestionar las actividades diarias de la cafetería, enfocándose en tres tipos de usuarios: **cliente**, **empleado** y **administrador**. Cada tipo de usuario tendrá funcionalidades específicas según su rol dentro de la organización.

El sistema tendrá una interfaz gráfica donde cada usuario podrá acceder a su respectivo panel de control y realizar las acciones correspondientes.

1. **Cliente:** Puede ver el menú actualizado, hacer pedidos, consultar el estado de su pedido y proporcionar retroalimentación sobre los productos, así como calificarlos.
2. **Empleado:** Gestiona los cobros de los pedidos que hacen los clientes. Puede registrar un pedido como pagado y confirmar la entrega.
3. **Administrador:** Tiene acceso completo para gestionar el menú, agregar o eliminar productos, ajustar precios, y generar reportes sobre el historial de pedidos y ventas. Además, puede revisar y gestionar las retroalimentaciones y calificaciones de los productos.

Supuestos y limitaciones

El sistema debe estar funcionando para el día 3 de Diciembre del 2024.

Entregables del proyecto

Los siguientes productos se producirán durante el proyecto:

- Diagrama de clases de la aplicación
- Código de la aplicación.
- Lista de todas las prácticas de construcción de software abordadas

Roles y responsabilidades

La siguiente tabla muestra las funciones representadas en el diagrama del proyecto y de sus responsabilidades primarias.

Nombre	Rol	Descripción
Russel Bonilla	Líder de Proyecto	Responsable de coordinar y supervisar todas las actividades del proyecto, asegurando el cumplimiento de plazos y objetivos.
Jaqueline Gongora	Especialista en Pruebas	Se encarga de realizar pruebas del sistema para identificar y documentar errores. Garantiza que el sistema cumpla con los requisitos.
Frida Pineda	Desarrollador de Funcionalidades del Sistema	Encargado de implementar las funcionalidades relacionadas con la gestión de usuarios y la lógica de las operaciones de cliente, empleado y administrador.
Victor Rosado	Desarrollador de Funcionalidades del Sistema	Encargado de implementar las funcionalidades relacionadas con la gestión de usuarios y la lógica de las operaciones de cliente, empleado y administrador.
Luisa Villanueva	Desarrollador de Interfaz Gráfica	Responsable de la creación y diseño de la interfaz gráfica del sistema, asegurando que las vistas de cliente, empleado y administrador sean intuitivas y responsivas.

Requisitos del sistema:

1. Gestión de Usuarios

- El sistema debe permitir el inicio de sesión para tres tipos de usuarios: cliente, empleado y administrador.
- El sistema debe tener un mecanismo de autenticación para validar a los usuarios por tipo (empleado, administrador).
- El sistema debe mostrar diferentes interfaces para cada usuario según sus permisos.

2. Funcionalidades del Cliente

- El cliente debe poder visualizar el menú de productos disponibles en la cafetería.
- El cliente debe poder seleccionar productos del menú para hacer un pedido.
- El cliente debe poder ver el total del pedido antes de confirmarlo.
- El sistema debe permitir al cliente confirmar y enviar el pedido a los empleados para su procesamiento.
- El cliente debe recibir una confirmación de que su pedido ha sido registrado exitosamente.
- El cliente debe poder proporcionar retroalimentación sobre los productos que ha pedido.
- El cliente debe poder calificar los productos en una escala de 1 a 5 estrellas.

3. Funcionalidades del Empleado

- El empleado debe poder ver todos los pedidos pendientes realizados por los clientes.
- El empleado debe poder registrar el cobro de cada pedido.
- El empleado debe poder marcar un pedido como entregado una vez se haya realizado el cobro.
- El empleado debe poder ver el estado de los pedidos (pendiente, entregado, pagado).

4. Funcionalidades del Administrador

- Repositorio de proyecto:** <https://github.com/soy-russ-bp/prueba1>

ID	Tareas realizables	Comienzo	Fin	Duración	oct-24		nov-24				dic-24		
					20-oct	24-oct	10-nov	12-nov	14-nov	19-nov	26-nov	01-dic	03-dic
1	Fase Inicial. <i>Definición de la Visión del Proyecto.</i> Establecer el propósito y los objetivos generales del sistema de administración para la cafetería, detallando los beneficios esperados para cada tipo de usuario <i>Definición de Requisitos del Sistema</i> Recopilar y documentar los requisitos funcionales y no funcionales <i>Creación del Plan de Desarrollo de Software</i> Establecer el cronograma y las fases de desarrollo, asignar recursos y definir puntos de control	20/10/2024	10/11/2024	24d									
2	Fase de Elaboración. Diseño base, diagramas de clases y prototipo de interfaz. Crear casos de uso detallados para cada funcionalidad del sistema, describiendo los pasos que cada tipo de usuario <i>Refinamiento de los Requisitos del Sistema</i> Revisar y afinar los requisitos identificados en la fase de inicio, asegurando que sean claros, específicos y estén alineados con los objetivos del proyecto <i>Diseño de la Arquitectura del Sistema</i> Definir la arquitectura del sistema, asegurando que sea modular, escalable y segura. (MVC) <i>Planificación de Pruebas para la Primera Iteración</i> Definir los criterios de prueba para la primera iteración de desarrollo, incluyendo pruebas unitarias, pruebas de integración y pruebas de usabilidad. <i>Diseño del Diagrama de Clases</i> Crear el diagrama de clases del sistema, detallando las clases principales, sus atributos, métodos y las relaciones entre ellas <i>Revisión y Aprobación de Diseño</i> Realizar una revisión de todos los documentos y diagramas de diseño creados	10/11/2024	14/11/2024	10d									
3	Fase de Construcción: Actividades y Entregables <i>Implementación del Módulo de Autenticación</i> Desarrollar el sistema de autenticación que permita a los usuarios iniciar sesión de acuerdo a su rol, asegurando un acceso seguro y restringido según los permisos de cada usuario. <i>Desarrollo de Funcionalidades para el Cliente</i> Descripción: Implementar las funcionalidades específicas para el rol de cliente, incluyendo la visualización del menú, la realización de pedidos y la consulta del estado de los pedidos. <i>Desarrollo de Funcionalidades para el Empleado</i> Desarrollar las funcionalidades para que los empleados puedan gestionar los cobros, registrar los pedidos como pagados y marcarlos como completados una vez entregados. <i>Desarrollo de Funcionalidades para el Administrador</i> Implementar las funcionalidades de administración, que incluyen la gestión del menú, el ajuste de precios y la generación de reportes sobre pedidos y ventas. <i>Integración de módulos.</i> Asegurar la integración adecuada entre los diferentes módulos para que interactúen correctamente y soporten el flujo de trabajo completo. <i>Pruebas Unitarias de Funcionalidades</i> Realizar pruebas unitarias para cada módulo y funcionalidad específica para verificar que cumplan con los requisitos y funcionen correctamente. <i>Pruebas de Integración</i> Probar la interacción entre módulos para asegurar que el sistema funcione de manera cohesiva. Verificar que la información fluya correctamente y que no haya fallos en los procesos intermodulares. <i>Revisión de Seguridad y Optimización del Código</i> Revisar el sistema para identificar posibles vulnerabilidades de seguridad y optimizar el código para mejorar el rendimiento	12/11/2024	02/12/2024	20d									
4	Fase de Transición <i>Pruebas de Desempeño y Seguridad</i> Ejecutar pruebas de rendimiento para evaluar el comportamiento del sistema bajo diferentes cargas de trabajo y verificar su estabilidad <i>Despliegue del Sistema</i> Implementar el sistema en el entorno de producción, migrando datos necesarios y configurando accesos para los usuarios finales.	01/12/2024	03/12/2024	3d									

Link del cronograma: [Cronograma.pdf](#)

Checklist de prácticas de construcción

Lineamiento de la asignatura	Cumple (Sí / No)
<i>Técnicas de construcción de variables y tipos de datos fundamentales</i>	
Inicializar cada variable cuando se declara.	
Idealmente, declarar y definir cada variable cerca de donde se usa.	
Inicializar los datos de los miembros de una clase en su constructor.	
Los nombres de las variables deben usar camelCase.	
Usar calificadores de cálculos en nombres de variables.	
Usar opuestos comunes en nombre de variables. Usar los opuestos con precisión.	
Nombramiento de índices de bucle. Si tiene que usar i, j y k, no usar para otra cosa que no sean índices de bucle.	
Si tiene varios bucles anidados, asignar nombres más largos a las variables de bucle para mejorar la legibilidad.	
Pensar en un nombre mejor que “flag” para las variables de estado. Una bandera nunca debe tener “flag” en su nombre.	
Utilizar nombres descriptivos para variables temporales en lugar de llamarlas como “temp” o “x”.	
Tener en cuenta los nombres booleanos típicos (done, error, found, success).	
Asignar nombres de variables booleanas que impliquen Verdadero o Falso.	

Utilizar nombres de variables booleanos positivos.	
Identificar constantes nombradas. Usar todas las letras mayúsculas, posiblemente con guiones bajos para separar las palabras, RECSMAX o RECS_ MAX.	
Diferenciar entre nombres de variables y nombres de rutinas.	
Identificar parámetros de solo entrada (<i>input-only</i>).	
Hacer que las conversiones de tipo sean obvias.	
Revisar el desbordamiento de enteros.	
Revisar si hay desbordamiento en resultados intermedios.	
Evitar cadenas y caracteres mágicos	
Estar atento a los errores off-by-one	
<i>Técnicas de organización de sentencias</i>	
Organizar el código para que las dependencias sean obvias.	
Nombrar las rutinas para que las dependencias sean obvias.	
Usar parámetros de rutina para hacer que las dependencias sean obvias.	
<i>Técnicas de construcción de estructuras de control de flujo</i>	
Escribir primero la ruta nominal a través del código; luego escribir los casos inusuales.	
Asegurarse de bifurcar correctamente en igualdad.	
Seguir la cláusula if con una declaración significativa.	
Poner los casos más comunes primero. If-then-else.	

Ordenar casos por frecuencia en las declaraciones case	
<i>Técnicas de construcción de procedimientos</i>	
Para nombrar una función, usar una descripción del valor devuelto. Por ejemplo, cos(), customerId.Next(), printer.IsReady0 y pen.CurrentColor().	
Para nombrar un procedimiento, utilizar un verbo fuerte seguido de un objeto. Por ejemplo: PrintDocument(), CalcMonthlyRevenues(), CheckOrderInfo() y RepaginateDocument).	
Poner las variables de estado o error al final en las rutinas.	
No utilizar parámetros de rutina como variables de trabajo.	
Usar parámetros con nombres significativos.	
<i>Técnicas de construcción de clases</i>	
Los nombres de las clases deben escribirse con PascalCase y deben ser sustantivos en singular.	
Los nombres de las interfaces deben escribirse con PascalCase.	
Preferir la herencia a la verificación extensiva de tipos.	
Si varias clases comparten un comportamiento común, pero no datos, derivarlas de una clase base común que defina las rutinas comunes.	
Si varias clases comparten datos y comportamientos comunes, heredar de una clase base común que defina las rutinas y los datos comunes.	
Heredar cuando desea que la clase base controle la interfaz; contener cuando desea controlar su interfaz.	
Hacer cumplir la propiedad singleton mediante el uso de un constructor privado.	

<i>Técnicas de documentación de código</i>	
Usar espacios en blanco para separar elementos de una sentencia no muy relacionados o mostrar precedencia de operadores.	
Uso de la sangría	
Escribir comentarios que describan la intención del código.	
Hacer que cada comentario cuente. No escribir más comentarios, hacer que el código sea más legible.	
Centrar los comentarios de los párrafos en el por qué en lugar del cómo.	