PROYECTO No. 01

ANALISIS DE SISTEMAS

Actualmente el procesamiento de órdenes se realiza de forma manual en donde el cajero anota el pedido del cliente en una nota, realiza la suma utilizando una calculadora y le notifica el monto total a pagar al cliente. Posterior a recibir el pago, el cajero procede a llenar la factura correspondiente. Al finalizar la transacción el cajero entrega una copia de la orden al área de cocina y otra copia al área de entrega. El encargado del área de entrega estará pendiente de cada orden y una vez recibida la comida desde el área de cocina se la entregará al cliente.

Los diferentes menús ofrecidos podrán ser personalizados, en número de opciones del menú, así como alternativas de cada menú, deberá contener una especificación de extras disponibles para cada uno de los tres tiempos (con costo y sin costo)

Usted como proveedor de software deberá realizar las tareas de análisis, diseño, planificación y programación necesarias para poner a funcionar la primera versión del sistema en un ambiente Web.

El sistema de información debe permitir efectuar transacciones básicas que incluyen los diferentes tiempos de servicio: desayuno, almuerzo y cena. Se debe crear la estructura para la personalización a los menús y será agregada de manera automática a los detalles del platillo. La sección de observaciones quedará reservada únicamente para pedidos extras.

REQUERIMIENTOS TECNICOS:

1. La aplicación deberá funcionar en un entorno Web en cualquiera de los navegadores
2. El modelo arquitectónico para utilizar será Modelo Vista Controlador -MVC- sobre un proyecto Java de Spring Boot. Para las capas de controladores y servicios intente separar las operaciones de sus componentes tomando en cuenta el concepto Separation of Concerns SoC.
3. Para la interfaz gráfica de usuario -GUI- deberá utilizarse HTML y CSS únicamente. Puede agregar Bootstrap o cualquier otra alternativa para la maquetación de las páginas.
4. El proyecto Java deberá utilizar plantillas como Thymeleaf para la elaboración de las páginas del sistema.
5. Los controladores deberán exponer operaciones utilizando cualquiera de los métodos HTTP GET, POST, PUT o DELETE

METODOLOGIA

1. Definir los requerimientos funcionales y no funcionales del sistema y colocarlos en una lista detallada.
2. Utilizar UML para realizar los modelos del sistema.
3. Diseñar sketch de las vistas
4. Crear y configurar proyecto SCRUM en Jira.
5. Crear las historias necesarias en el backlog, configurar sprints de 3 días.
6. Agregar descripción, criterios de aceptación, bosquejos e información adicional a cada historia antes de ser asignado.
7. Utilizar una rama Git diferente para cada feature
8. Actualizar los comentarios de cada historia cada día.
9. Cada integrante hace el merge de sus features al finalizar cada sprint
   1. La pestaña Insights > Network en GitHub debe reflejar cada acción Git efectuada por los integrantes del equipo. Recuerde NO eliminar las ramas.

CONSIDERACIONES

1. Diagrama de casos de uso. No se requieren especificaciones.
2. 2. Diagrama de secuencias en UML -para un proceso-, este diagrama deberá reflejar mensajes y respuestas acordes al código fuente generado.
3. Diagrama de clases en UML 2
4. Código fuente en lenguaje Java 8+ con MAVEN
5. Utilizar únicamente capas para controladores, servicios y entidades. No hay capa de acceso a datos dado que no se requiere repositorio.
6. Elaboración de plantillas HTML para las páginas con Thymeleaf o Freemarker
7. Proyecto en Jira con la administración del proyecto
8. Repositorio GitHub privado con las ramas correspondientes a cada feature.
9. Agregar como colaborador al catedrático por medio de su cuenta glunagt para poder calificar a partir de la rama master.
10. Fecha límite de entrega:
11. El proyecto deberá trabajarse en equipos de 4 personas máximo. Las tareas deberán distribuirse de la siguiente manera.
    1. Project Manager
    2. Analista y Diseñador
    3. Desarrolladores -2-