

Tarea Corta #3-Prototipo

Instituto Tecnológico de Costa Rica
Escuela Ingeniería en Computadores
Bases de Datos (CE3101)
Primer Semestre 2024



Objetivo General

- Desarrollar **Prototipo** para una aplicación que permita gestionar la descripción del caso.

Objetivos Específicos

- Crear un API/WebService/REST Service (C#).
- Crear una página web.
- Usar herramientas como Angular/React, Bootstrap, HTML5, CSS3, Crystal Report o Reporting Services.
- Instalar localmente una aplicación Web (Front End y Back End).

Descripción de la tarea corta

El TEC desea mejorar la gestión del comedor institucional, especialmente las filas para poder hacer el uso del mismo.

Dado lo anterior se ha tomado la decisión de implementar el **proyecto RestTEC** con el objetivo de desarrollar un sistema que permita **registrar y gestionar los pedidos** de los usuarios. A continuación, se describen los requerimientos de software asociados a esta tarea corta:

- **Aplicación Web.** Debe ser desarrollada utilizando Angular/React, Bootstrap, CSS y HTML5. Este sitio se compone de varias vistas las cuales se describen a continuación:
 - ◆ **Vista Administración.** En esta vista se podrá:
 - *Login:* Al iniciar sesión con su correo **electrónico y password** el sistema le mostrará por defecto **todos los pedidos que estén activos** en el sistema.
 - *Gestión de tipos de platos:* Las operaciones que debe proveer esta vista son **creación, actualización y eliminación de opciones del menú**. Para **cada tipo de plato solamente es necesario el nombre y una descripción**.
 - *Gestión del menú:* Las operaciones que debe proveer esta vista son **creación, actualización y eliminación** de opciones del menú. Para cada opción de menú es importante contar con la siguiente información: **Nombre del plato, precio, cantidad de calorías, tipo**.
 - **Vista de Reportes.** Permitirá a la gerencia analizar:
 - Top 10 de los platos más vendidos.
 - Top 10 de los platos que más ganancias generan.
 - Top 10 de los platos con mejor feedback

- Top 10 de los clientes que más órdenes han generado.

◆ Vista del Chef. En esta vista se podrá:

- *Login*: Al iniciar sesión con su correo electrónico y password el sistema le mostrará por defecto todos los pedidos que estén activos en el sistema y que hayan sido tomados por el chef.
- *Tomar pedidos*: Esta funcionalidad debe permitir a los chefs visualizar los pedidos que no han sido asignados y el chef podrá seleccionar un pedido para preparar.
- *Control de pedidos*: Esta funcionalidad debe permitir a cada chef visualizar los pedidos que tiene asignados y mostrar el tiempo restante que tiene para completar cada plato del pedido (tiempo transcurrido versus tiempo estimado de preparación del plato). Al completarse el tiempo de un plato el sistema debe esperar que el chef confirme que se completó, para que el cliente pueda observar que el plato ha sido completado.
- *Reasignar un pedido*: Un chef tiene la posibilidad de visualizar los pedidos que han sido tomados por otros chef y tomar el control de este pedido. También tiene la posibilidad de tomar control de un pedido que haya sido asignado a otro chef, para lo cual ambos deben confirmar la transacción (Si hubiera back-end en este caso con solo la confirmación del primer chef es suficiente).

→ **App Móvil.** Esta es una aplicación móvil que permite a los clientes interactuar con el sistema. El objetivo de esta aplicación es que el cliente use su Smartphone para realizar los pedidos sin necesidad de hacer la fila en el comedor.

- ◆ *Registrarse*: Las operaciones que debe proveer esta vista son creación, actualización y eliminación de cuentas. Los principales datos de los clientes son: número de cédula, nombre, apellidos, dirección (provincia, cantón, distrito), fecha de nacimiento, teléfonos, correo electrónico y password.
- ◆ *Login*: Al iniciar sesión con su correo electrónico y password el sistema le mostrará el menú con las opciones que se encuentren activos en el sistema.
- ◆ *Administrar el carrito de compras*: Esta vista habilita al cliente para que agregue platos al carrito de compras y una vez que esté listo los revise antes de confirmarlo. El cliente puede eliminar platos o modificar las cantidades solicitadas. Este carrito mostrará una línea por cada producto y una total de monto a pagar.
- ◆ *Generar pedidos*: Una vez confirmado el carrito de compras el sistema debe proveer la confirmación del pedido y mostrar al cliente la factura. Para cada factura es necesario establecer la fecha y hora en que se generó, platos incluidos en el pedido, monto total del pedido y un consecutivo específico.
- ◆ *Visualizar estado del pedido*: Esta funcionalidad debe permitir a cada cliente visualizar los pedidos activos y mostrar el porcentaje restante para que sea completado cada plato del pedido (porcentaje de completitud calculado considerando el tiempo transcurrido con respecto al tiempo de preparación estimado).

- ◆ *Feedback*: Una vez que se haya realizado la compra y le sea entregada al cliente, este podrá calificar el pedido. Para esto es necesario que se registre el usuario que brinda el feedback, la fecha y hora en que se realizó y la calificación de uno a cinco estrellas.
- **Seguridad**. Dado que existirán varios clientes los pedidos de un cliente no pueden ser vistos por otros clientes.
- **Implementación**. Para la implementación debe considerar:
 - ◆ Toda la implementación será de forma local (localhost). No se permitirá el uso de servicios en la nube como AWS o Azure.
 - ◆ No está permitido el uso de tecnologías aspx.
 - ◆ Lenguaje C# y .NET
 - ◆ Deben utilizar alguna herramienta para el control de versiones del código e incluir los links de los repositorios en el documento. Se recomienda el uso de GitHub. El repositorio debe ser público o brindar acceso a los correos mrivertec@gmail.com

Algunas consideraciones adicionales:

- Cada usuario debe registrar al menos un número telefónico donde pueda ser localizado.
- Cada plato tiene una serie de ingredientes que son necesarios para su preparación.
- Cada plato tiene un tiempo estimado de preparación que es calculado en el momento en que se genera el pedido y es calculado considerando la cantidad de ingredientes que tiene el plato y la cantidad de pedidos que tiene asignado el chef.
- Para cada plato debe calcularse una hora estimada de finalización, el cual es calculado usando la hora en que se generó el pedido y el tiempo estimado de preparación del pedido.
- La Base de Datos debe ser un archivo XML, Jason, Txt.

Entregables

A continuación, se listan los entregables de la tarea:

- Código fuente documentado.
- Manual de Usuario (PDF - para el usuario final de la aplicación).
- Documento de instalación (PDF - para técnicos que instalan las aplicaciones desarrolladas).
- Documentación técnica del proyecto (PDF – para el equipo de soporte)
 - ◆ Descripción de los métodos implementados.
 - ◆ Descripción de las estructuras de datos desarrolladas.
 - ◆ Descripción detallada de los algoritmos desarrollados.
 - ◆ Problemas conocidos: Corresponde a cualquier problema que no se ha podido solucionar en el trabajo.
 - ◆ Problemas encontrados: Descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
 - ◆ Documentación que evidencie el trabajo en equipo:
 - Actividades planeadas, su responsable y fecha de entrega (Plan de trabajo).
 - Minutas de sesiones de trabajo (Seguimiento al plan de trabajo).

- Actividades realizadas por cada estudiante. Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, entre otros. Se debe describir todo por más insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo, llevan seguimiento de todo en el tiempo. Imaginen que, si un compañero los releva en su trabajo, le bastaría con leer sus bitácoras para seguir el trabajo.
- ◆ Conclusiones y recomendaciones del proyecto.
- ◆ Bibliografía consultada para el proyecto.
- Aplicación WEB con reportes.
- Aplicación Móvil.
- Web API/Service/REST Service.

Aspectos operativos y evaluación:

1. **Fecha de entrega:** 20 de Marzo del 2024.
2. **Valor:** 6%
3. **Nombre código:** RestTEC
4. El trabajo es **en grupos de 5 personas**.
5. Debe hacer entrega de todos los documentos solicitados y los links de acceso a los repositorios.
6. Es obligatorio integrar toda la solución.
7. Las funcionalidades implementadas corresponderán a un valor de 70%, la documentación 20% y la defensa 10%. Cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, dado que se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
8. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código, Documentación y defensa.
9. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que, aunque el proyecto esté 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, uso de herramientas solicitadas, calidad de documentación interna, externa y trabajo en equipo.
10. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
11. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
12. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
13. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - a. Si no se entrega documentación, automáticamente se obtiene una nota de cero en la tarea corta.
 - b. Si no se utiliza un manejador de código se obtiene una nota de cero en la tarea corta.

- c. Si no se entrega el punto 6 de la documentación se obtiene una nota de cero en la tarea corta.
 - d. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de cero en la tarea corta.
 - e. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - f. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de cero en la tarea corta.
 - g. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de cero en la tarea corta.
 - h. Si no se presenta a la revisión, automáticamente se obtiene una nota de cero en la tarea corta.
14. Cada grupo tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
 15. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
 16. Cada grupo es responsable de contar con los equipos requeridos para la revisión. Si es necesario que el profesor coordine para contar con algún equipo debe comunicarlo con anticipación al profesor.
 17. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
 18. Las revisiones se realizan con los estudiantes matriculados en el curso. Cualquier persona fuera de estos y los mencionados en el punto 17, no pueden participar en la revisión.
 19. Después de enviada la nota final del proyecto el estudiante tendrá un máximo de 3 días hábiles para presentar un reclamo siempre y cuando la funcionalidad esté completa.

Referencias

AngularJS (2019-02-05). Recuperado de: <https://angular.io/>

Bootstrap Themes & Templates (2019-02-05). Recuperado de: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>

How to Write Doc Comments for the Javadoc Tool. (2022-02-25). Recuperado de: <http://www.oracle.com/technetwork/articles/java/index-137868.html>

C# Coding Conventions (C# Programming Guide). (2022-02-25). Recuperado de: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>

Anexos y estructuras (Ministerio de Hacienda). (2022-02-25). Recuperado de: <https://www.hacienda.go.cr/ATV/ComprobanteElectronico/frmAnexosyEstructuras.aspx#>