

Tarea Corta #1

Instituto Tecnológico de Costa Rica
Área Académica Ingeniería en Computadores
Bases de Datos (CE3101)
Segundo Semestre 2021



Objetivo General

- Desarrollar 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, Bootstrap, HTML5, CSS3, Crystal Report o Reporting Services.
- Instalar localmente una aplicación Web (Front End y Back End).

Descripción de la tarea corta

Tec Bank es un banco global con presencia en más de 70 países y ha incursionado en nuestro país, la junta directiva del banco ha decidido: como una de las aristas de su nueva estrategia de mercado realizar la mayor recaudación de fondos posible, para lo cual el banco está implementando el procesamiento de pago de tarjetas con pequeñas empresas, la primer prueba se realizara con el minisúper "Las 3 Marías" L3M y la sucursal central del banco con sede en Cartago y debe entrar a regir el 8/Sep/2021 a primera hora esto por un acuerdo político con el ministro de la Presidencia, cualquier retraso por cada semana se multará a la empresa desarrolladora con una multa de \$50,000 semanales (estimados de la siguiente manera \$20,000 por perdidas al minisúper y \$30,000 por pérdida de credibilidad del banco y el gobierno).

Para poder hacerle frente a la nueva estrategia el banco se ve en la urgida necesidad de actualizar parte de su core financiero de manera que sea más ágil el proceso normal de administración bancario que le permita liberar recurso humano para que este recurso se ocupe de la nueva operación del banco, a continuación, se describen los principales requerimientos a desarrollar por su empresa.

→ **Aplicación Web.** Debe ser desarrollada utilizando Angular, Bootstrap, CSS y HTML5. Este sitio se compone de varias vistas las cuales se describen a continuación:

- ◆ Vista Administración. En esta vista se podrá:
 - ◆ Gestión de roles: El sistema debe permitir ingresar, modificar y eliminar roles, los cuales solamente almacenaran un nombre y descripción.
 - ◆ Gestión de Clientes: El sistema debe permitir ingresar, modificar y eliminar clientes, Se debe almacenar: nombre completo, cédula, dirección, teléfono, ingreso mensual y tipo de cliente (Físico, Jurídico).
 - ◆ Gestión de cuentas: El sistema debe permitir ingresar, modificar y eliminar Cuentas, de las cuales se debe almacenar: número de cuenta, descripción de la cuenta, moneda (Colones, Dólares, Euros), tipo de cuenta (Ahorros, Corriente) y el cliente al que

pertenece.

Adicionalmente el sistema debe proveer la forma de realizar un depósito o retiro de efectivo a una cuenta.

- ◆ **Gestión de Tarjetas:** El sistema debe permitir ingresar y modificar tarjetas, de las cuales se guarda el número de tarjeta, tipo de tarjeta (Débito o crédito), fecha de expiración, código de seguridad y saldo disponible/monto de crédito disponible.
 - **Tarjetas de Débito:** En caso de las tarjetas de débito se debe poder eliminar tarjetas, estas estarán asociadas únicamente a una cuenta.
 - **Tarjetas de Crédito:** En caso de las tarjetas de crédito se debe poder eliminar tarjetas siempre y cuando no tengan saldos pendientes.
- ◆ **Gestión de Asesores de Crédito:** Los asesores de crédito son empleados del banco que ofrecen los préstamos a los clientes, de ellos se almacena el nombre completo, número de cédula, fecha de nacimiento. Ellos tienen metas de ventas tanto en colones como en dólares, una vez cumplidas las metas ellos reciben una comisión de un 3% sobre el monto de los créditos otorgados en ese mes. Por tanto, es necesario un reporte que se emite todos los primeros de cada mes el cual muestra el nombre del asesor y los siguientes datos resumidos: monto de la meta en colones y en dólares, total de los créditos en colones y en dólares y las comisiones ganadas tanto en colones como en dólares.
- ◆ **Gestión de Préstamos:** El sistema debe permitir ingresar préstamos, así como realizar cálculo de pagos (una vez otorgado el préstamo, se debe generar un calendario de pagos donde se especifique el monto de amortización e intereses el interés es sobre saldo), pagos, pagos extraordinarios (los cuales no pagan intereses), y recálculo de pagos (se ejecuta inmediatamente después del pago extraordinario). De los préstamos se debe almacenar el monto original del préstamo, el saldo, el cliente que solicitó el préstamo, tasa de interés y los pagos de cada uno de los préstamos.
- ◆ **Gestión de Mora:** Este será un **reporte** que se genera todos los cinco de cada mes automáticamente y cuando el usuario lo desee, este mostrará el nombre completo del cliente, cédula, número de préstamo, cuotas vencidas, monto adeudado.
- ◆ **Aplicación Web Cliente.** Esta será una página donde al ingresar el cliente tendrá acceso a todos los productos que tiene con el banco (cuentas, tarjetas crédito y préstamos)
 - **Cuentas**
 - **Ver movimientos de mis cuentas:** el sistema debe proveer un detalle de todos los movimientos que se han realizado a una cuenta (Depósitos-Retiros). Así mismo podrá observar las **tarjetas de débito** asociadas a la cuenta.
 - **Realizar transferencia de dinero a otra cuenta.** El sistema debe proveer la opción de realizar una transferencia de efectivo a otra cuenta.
 - **Tarjetas**
 - **Pago de Tarjetas:** El sistema debe proveer la funcionalidad para poder realizar un pago a una tarjeta de crédito.
 - **Listado de compras realizadas:** Dado un rango de fechas el sistema debe proveer la lista de compras realizadas con esa tarjeta.

- *Prestamos*
 - *El sistema debe proveer la funcionalidad para poder realizar el pago normal o extraordinario a un préstamo.*
- *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 aplicación debe de proveer la siguiente funcionalidad tal como se detalla en la Aplicación Web Cliente.

→ **Seguridad.** Los datos de un cliente solo pueden ser vistos por él, ningún otro cliente podrá ver los datos.

→ **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 y kevin.gonzalezs0795@gmail.com

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 relea 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:** De acuerdo a lo establecido en el TEC Digital (15-Sep-2021).
2. **Valor:** 6%
3. **Nombre código:** TEC Bank
4. El trabajo es **en grupos de Cuatro 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/>