

1. Objetivo General

Desarrollar una aplicación que permita manejar la descripción del caso expuesto en el punto 4.

2. Objetivos Específicos

- 2.1. Crear un WebService/REST Service/API (C#).
- 2.2. Crear una página Web.
- 2.3. Usar herramientas como Angular, Bootstrap, HTML5, CSS3, Cristal Report o Reporting Services.
- 2.4. Instalar localmente una aplicación Web (Front End and Back End).

3. Datos Generales

- 3.1. El valor del proyecto: 5-10%
- 3.2. **Nombre código: L3M Super**
- 3.3. La tarea debe ser implementada grupos de 4 personas.
- 3.4. La **fecha de entrega:** 16-Ago-2019
- 3.5. Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo al reglamento.

4. Descripción del caso de estudio

La junta directiva del supermercado “Las 3 Marías” ha decidido que es el momento indicado para empezar una expansión de su negocio.

Dentro del análisis FODA que realizaron para iniciar su expansión encontraron como la debilidad más grande la forma en que es administrado el supermercado, actualmente una de las socias lleva todos los registros de los proveedores, compras, ventas, empleados y planilla en una hoja de cálculo. El análisis demostró que por cada supermercado que se abra se requeriría la contratación de un recurso para realizar estas labores.

El contratar un recurso implica muchos riesgos, desde que se ausente por cualquiera de los motivos que sea, hasta que este recurso eventualmente deje la empresa y cree su propio supermercado creando competencia a L3M.

Con este escenario la directiva ha decidido que es mucho mejor licitar la creación de un sistema de software que facilite la administración del supermercado.

El detalle de los requerimientos del software se encuentra expuesto en la sección 4.1, como este proceso de expansión ya ha empezado y el sistema de administración se hace indispensable en caso de retraso por cada semana se multará a la empresa desarrolladora con una multa de \$20000 semanales (costo de mantener cerrado el punto de venta actual y el que se planea abrir).

4.1. Requerimientos del Software

Aplicación Web: Debe ser desarrollada utilizando Angular, Bootstrap, CSS y HTML5. Este sitio se compone de diferentes vistas.

4.1.1.1. **Vista Administración:** en esta vista se podrá:

- 4.1.1.1.1. Gestión de roles: El sistema debe permitir ingresar, modificar y eliminar roles, los cuales solamente almacenaran un nombre y descripción.
- 4.1.1.1.2. Gestión de sucursales: El sistema debe permitir ingresar, modificar y eliminar sucursales, las cuales almacenaran un nombre, dirección, teléfono, administrador.
- 4.1.1.1.3. Gestión de trabajadores: El sistema debe permitir ingresar, modificar y eliminar trabajadores, de los cuales se guarda cedula, nombre completo, fecha de nacimiento, fecha de ingreso al supermercado, sucursal donde labora y el salario por hora del empleado.
- 4.1.1.1.4. Gestión de Proveedores: El sistema debe permitir ingresar, modificar y eliminar proveedores, de los cuales se guarda nombre completo, cedula (Física o Jurídica).
- 4.1.1.1.5. Gestión de Productos: El sistema debe permitir ingresar, modificar y eliminar productos, de los cuales se guarda nombre, descripción, código de barras, proveedor, precio de compra, si paga impuesto o no, si tiene descuento o no; el sistema deberá permitir la flexibilidad de asignar un precio por sucursal.
- 4.1.1.1.6. Gestión de Compras: El sistema debe permitir ingresar, modificar y eliminar compras, de los cuales se guarda una descripción, fecha real de la compra, fecha de registro de la compra, proveedor, foto de la compra y sucursal que registra la compra.
- 4.1.1.1.7. Asignación de horas laboradas por trabajador. En esta vista el administrador podrá indicar cuantas horas laboro un empleado en una sucursal en una semana específica, en el sistema se debe permitir indicar cuantas horas extra laboro el trabajador.
- 4.1.1.1.8. Generación de planilla. Este proceso se ejecuta solo si se ha registrado horas para una semana de lo contrario debe sugerir al administrador realizar el proceso de asignación de horas laboradas por trabajador para la sucursal. Si los datos están correctos se genera un reporte de todos los trabajadores con la cantidad de horas laboradas detallando cuantas fueron horas ordinarias y cuantas fueron horas extra. Una hora extra equivale a una hora ordinaria por 1.5

4.1.1.2. **Vista de Reportes.**

- 4.1.1.2.1. Listado de productos: este reporte mostrara todos los productos de la cadena de supermercados, el costo del producto y la existencia en toda la cadena.
- 4.1.1.2.2. Listado de productos por sucursal: este reporte mostrara ordenado por nombre todos los productos que ofrece una sucursal en específico, la descripción

del producto, el precio y la existencia en esa sucursal.

4.1.1.2.3. Listado de gastos: este reporte tendrá como entrada una sucursal específica, y un rango de fechas y mostrará todos los gastos de esa sucursal en las fechas mencionadas, donde se mostrará la fecha del gasto, el proveedor, descripción del gasto y monto del gasto, ordenados por la fecha del gasto. En caso de que la sucursal no se detalle como entrada para el reporte se deberá mostrar todos los gastos de todas las sucursales.

4.1.1.3. **Vista del Punto de Venta.**

4.1.1.3.1. En esta vista se podrá realizar la facturación de productos por cada venta se almacenará entre otros datos: el cajero, sucursal, producto(s) que se vende(n), cantidad, precio unitario por producto y precio total, finalmente se podrá ver el precio de la venta total, descuento e impuesto.

App Móvil.

Esta aplicación debe de proveer la siguiente funcionalidad tal como se detalla en la página web.

- Gestión de trabajadores
- Gestión de Compras
- Asignación de horas laboradas por trabajador

Core L3M.

Este módulo representa el cerebro de los demás sistemas que se conectaran a él para poder realizar las principales operaciones de un modo centralizado. Este módulo será un conjunto de servicios REST que serán consumidos por la aplicación web y aplicación móvil.

Seguridad.

Los datos de ventas de una sucursal solo pueden ser observados por los empleados de esa sucursal.

Implementación.

Toda la implementación será de forma local IIS y SQL Server, no se permitirá servicios en la nube como AWS o Azure.

5. Entregables

- 5.1. Manual de Usuario.
- 5.2. Documentación Técnica y del proyecto (descrita en el punto 6).
- 5.3. Documento de instalación.
- 5.4. Aplicación WEB.
- 5.5. Aplicación Móvil.
- 5.6. Web Service/REST Service/API.

6. Documentación

- 6.1. Se deberá documentar el código fuente.
- 6.2. Se deberá entregar un documento que contenga:
 - 6.2.1.Descripción de los métodos implementados.
 - 6.2.2.Descripción de las estructuras de datos (entidades) desarrolladas.
 - 6.2.3.Descripción detallada de los algoritmos desarrollados.
 - 6.2.4.Problemas sin solución: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
 - 6.2.5.Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
 - 6.2.6.Documentación de evidencia del trabajo en equipo.
 - 6.2.6.1. Actividades planeadas y su responsable. (Plan de trabajo)
 - 6.2.6.2. Evidencia de uso de un manejador de código (se recomienda Github).
 - 6.2.7.Conclusiones y Recomendaciones del proyecto.
 - 6.2.8.Bibliografía consultada en todo el proyecto
- 6.3. Diagrama de clases y un documento que explique el porqué del diseño.
- 6.4. Diagrama entidad-relación de la Base de Datos.

7. Evaluación

1. El proyecto tendrá un valor de un 80% de la nota final, debe estar funcional.
2. La documentación tendrá un valor de un 20% de la nota final, cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
3. La defensa tendrá su propia evaluación.
4. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código y Documentación.
5. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del paradigma OOP, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.
6. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
7. 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.
8. De las notas mencionadas en los puntos 1 y 2 se calculará la Nota preliminar y la nota Final del Proyecto será la nota preliminar * la nota de la defensa.
9. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
10. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - 10.1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
 - 10.2. Si no se utiliza un manejador de código se obtiene una nota de 0.

- 10.3. Si no se entrega el punto 4 de la documentación se obtiene una nota de 0.
- 10.4. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
- 10.5. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
- 10.6. 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 0.
- 10.7. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de 0.
- 10.8. El no presentarse a la revisión se obtiene una nota de 0.
- 11. 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.
- 12. 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.
- 13. Cada grupo es responsable de llevar los equipos requeridos para la revisión.
- 14. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
- 15. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 14, no pueden participar en la revisión.
- 16. 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.

8. 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/>