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

2. **Objetivos Específicos**
   2.1. Aplicar los conceptos del modelo conceptual y relacional.
   2.2. Crear una Base de Datos en MS Server.
   2.3. Crear un WebService/REST Service.
   2.4. Crear una página Web.
   2.5. Usar herramientas como Angular, Bootstrap, HTML5, CSS, Entity Framework, y reporting services o cristal reports.
   2.6. Crear un documento de instalación.
   2.7. Crear un plan de proyecto.

3. **Datos Generales**
   3.1. El valor del proyecto: 25%
   3.2. Nombre código: TECres
   3.3. La tarea debe ser implementada grupos de 4 personas.
   3.4. La **fecha de entrega:**
      3.4.1. Plan de proyecto: 17/ Setiembre /2019
      3.4.2. Resumen Ejecutivo Avance 1: 19/ Setiembre /2019
      3.4.3. Resumen Ejecutivo Avance 2: 26/ Setiembre /2019
      3.4.4. Resumen Ejecutivo Avance 3: 3/ Octubre /2019
      3.4.5. Funcionalidad completa: 10/ Octubre /2019.
      3.4.6. **Todos los 'subject' del envió debe ser: CE3101_P1_PT o CE3101_P1_RE[1-4] y deben enviarse al correo mriveramTec@gmail.com.**
   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**
   TECres está conformado por cientos de agentes inmobiliarios calificados que comparten y administran miles de propiedades y clientes en una base de datos centralizada. El objetivo principal es proveer una colaboración para que la venta o alquiler de propiedades se realice lo más pronto posible. La herramienta consta de tres vistas:

Vista Público en General: esta es la plataforma que permitirá al público en general ver todas las propiedades que se encuentra registradas en el sitio para venta o alquiler. Para ingresar a buscar propiedades no es necesario tener un usuario.
Vista Cliente: esta plataforma permite a los dueños registrar sus propiedades, para posteriormente crear un anuncio, una vez finalizado su anuncio podrá ver las estadísticas. En este caso se requiere una autenticación del usuario.
Vista Administrador: Ésta es la plataforma que permite a los dueños del sitio administrar su negocio, eso significa realizar toda la configuración del sistema como la facturación a los clientes.

4.1. **Requerimientos del Software**

4.1.1. **Público en General.**

4.1.1.1. *Buscar Propiedades:* el sistema debe permitir a cualquier usuario buscar una propiedad con los siguientes filtros: Ubicación, cantidad de habitaciones, precio mínimo y precio máximo. El resultado sería un listado de todas las propiedades que cumplan con los criterios de búsqueda, este listado mostrará la foto principal de la propiedad, ubicación exacta, precio, metros cuadrados de construcción y del terreno y el precio resaltado. Si el usuario desea ver mas detalles puede dar click en el título o foto del anuncio y se mostraran todos los detalles de la propiedad con un carrusel con las demás fotos secundarias.

4.1.1.2. *Registro de interesado/comprador*: El sistema debe permitir la creación, modificación y eliminación de interesados. Se conoce del interesado el nombre y apellidos, genero, fecha de nacimiento, lugar de residencia (ubicación), ingresos por mes y ocupación.

4.1.1.3. *Enviar Mensajes/Notificaciones:* El sistema debe proveer una forma fácil de comunicarse con el dueño y vendedor de la propiedad, para lo cual podrá crear un mensaje. Se podrá enviar mensajes siempre y cuando este registrado como un comprador.

4.1.2.**Cliente.**

4.1.2.1. *Gestión de Clientes:* El sistema debe proveer la posibilidad de crear, modificar y eliminar clientes. Para el registro de clientes se requiere conocer, número de cédula, nombre y apellidos, nacionalidad, perfil de cliente (Agente, constructor, propietario, anunciante), correo electrónico y usuario y password.

4.1.2.2. *Gestión de Propiedades:* El sistema debe permitir crear, modificar y eliminar propiedades. Para el registro de una propiedad se desea conocer: Título de la propiedad, descripción de la propiedad, ~~dueño (cliente)~~, Tipo Inmueble (Lote, casa, apartamento), ubicación de la propiedad, metros cuadrados del terreno, metros cuadrados de construcción, cantidad de habitaciones, cantidad de baños, cantidad de parqueos, tipo de piso (concreto lujado, cerámica, porcelanato), niveles, gimnasio, piscina, parqueo para visitas, listado de fotos del inmueble y **precio.**

4.1.2.3. *Gestión de Anuncios:* El sistema debe proveer la posibilidad de crear, modificar o eliminar anuncios. Todo anuncio debe estar asociado a una propiedad. Para el

Commented [RM(GC1]: Como ya se está registrado no es necesario solicitarlo.

registro de un anuncio se debe conocer: Tipo de Anuncio (Normal, destacado, oro, platino), el público meta (definido por el usuario), si es un alquiler o una venta, fecha inicio y fin del anuncio y tarjeta de crédito donde realizar el cargo.

4.1.2.4. *Gestión de Público:* El Sistema debe permitirle al cliente poder crear, modificar y eliminar un público al cual están dirigidos. Para la creación de un público se requiere saber a quienes desea enfocarse, Hombres/Mujeres, Rango de edades, ubicación, ingresos.

4.1.3.**Administrador**

4.1.3.1. *Gestión de Agente Vendedor*: El sistema permitirá la creación, modificación y eliminación de vendedores. Cuando se crea un vendedor se desea saber, nombre y apellidos, número de cédula, fecha de ingreso.

4.1.3.2. *Gestión de Tipos de Inmuebles*: El sistema debe permitir la creación, modificación y eliminación. Los tipos predefinidos para Tipos de Inmueble son: Lote, Casa y Apartamento.

4.1.3.3. *Gestión de Ubicaciones:* Inicialmente el sistema esta enfocado en Costa Rica por lo que se debe permitir crear, modificar y eliminar las ubicaciones, estas al menos deben de tener provincia, cantón y distrito.

4.1.3.4. *Gestión de Tipos de anuncio:* El sistema debe permitir crear, modificar y eliminar los tipos de anuncio. Al crear un tipo de anuncio se desea conocer un nombre, una descripción y el costo por día de ese tipo. Los tipos predefinidos son Normal (1000), Destacado(2000), Oro(3000), Platino (4000).

4.1.3.5. *Gestión de Perfil de Clientes:* El sistema debe permitir crear, modificar y eliminar los perfiles de clientes. Al crear un perfil se desea conocer un nombre y una descripción. Los perfiles predefinidos son: Agente vendedor, constructor, propietario, anunciante.

4.1.3.6. *Gestión de ocupaciones:* El sistema debe permitir crear, modificar y eliminar las ocupaciones. Se mantendrá un nombre de ocupación.

4.1.3.7. *Gestión de administradores:* El sistema debe proveer la opción para poder crear administradores que son quienes administran TECres. Del administrador se conoce el nombre y apellidos, cédula, fecha de nacimiento, fecha de ingreso a TECres.

4.1.3.8. Aprobación de Anuncios: Una vez el usuario crea un anuncio este es enviado para su aprobación. Un administrador de la TECres revisa manualmente que las fotografías realmente sean de la propiedad, que la tarjeta de crédito sea válida, realiza el cargo a la tarjeta y posteriormente aprueba el anuncio.

4.1.3.9. Asignación de Vendedor a un anuncio/propiedad. Un administrador de TECres es el encargado de asignar un vendedor de la compañía a las propiedades con el fin de agilizar y darle seguimiento a la propiedad y sus anuncios.

4.1.3.10. Generación de Estadísticas. Cada vez que se crea un anuncio se desea ver la cantidad de visitas que ha tenido además de los mensajes enviados.

4.1.3.11. Facturación. Cada vez que se genera un anuncio se crea una pre-factura con el monto del anuncio (cantidad de días * tipo de anuncio). La factura queda en firme una vez el anuncio es aprobado y esta es enviada por correo al cliente.

4.1.3.12. **Reportes.**

4.1.3.12.1. Reporte de ventas. El objetivo de este reporte es ver el detalle de todas las facturas realizadas en un periodo de tiempo definido por el usuario. El reporte mostrara el nombre del cliente, la propiedad y el monto facturado.

4.1.3.12.2. Reporte de Anuncios por vencer. En este reporte se desea observar el nombre del cliente, la propiedad, nombre del vendedor y cuando vence el anuncio.

4.1.3.12.3. Reporte de comisiones. El objetivo de este reporte es realizar el pago de comisiones a los vendedores. Por cada anuncio realizado a una propiedad de un vendedor este obtendrá un 5% por comisión. El reporte deberá estar agrupado por vendedor y debe mostrar el anuncio, monto del anuncio y la comisión de ese anuncio junto con el total ganado por el vendedor.

4.1. **Requerimientos NO funcionales del Sistema**

- El Sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS y Entity Framework).
- La Base de Datos debe estar en MS Server.
- La capa de servicios debe estar desarrollada en C# y debe ser desplegada en el IIS.
- La Base de Datos debe estar al menos en tercera forma normal.
- Se deben implementar al menos 5 procedimientos almacenados y 2 triggers.
- El equipo de trabajo debe seleccionar a uno de sus miembros como único punto de contacto. Todas las comunicaciones y solicitudes deben ser a través de dicho punto de contacto.

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. Plan de Proyecto.
5.5. Script de Base de Datos.
5.6. Script de población de Base de Datos.
5.7. Aplicación WEB.
5.8. Web Service/REST Service.
5.9. Minutas

6. **Documentación**

6.1. Se deberá documentar el código fuente.
6.2. Se deberá entregar un documento que contenga:
6.2.1. Modelo conceptual utilizando la notación de Chen.
6.2.2. Modelo relacional.
6.2.3. Descripción de las estructuras de datos desarrolladas (Tablas).

6.2.4. Descripción de los Store Procedures/Funciones/triggers implementados.
6.2.5. Descripción detallada de la arquitectura desarrollada.
6.2.6. Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
6.2.7. Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
6.2.8. Documentación de evidencia del trabajo en equipo.
6.2.8.1. Actividades planeadas, su responsable y fecha de entrega. (Plan de trabajo)
6.2.8.2. Minutas de sesiones de trabajo. (Seguimiento al plan de trabajo)
6.2.8.3. Actividades realizadas por cada estudiante. (Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. 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.
6.2.8.4. Evidencia de uso de un manejador de código (se recomienda Gitgub).
6.2.9. Conclusiones y Recomendaciones del proyecto.
6.2.10. Bibliografía consultada en todo el proyecto
6.3. Diagrama de clases y un documento que explique el porqué del diseño.

7. **Evaluación**
1. El proyecto tendrá un valor de un 65% de la nota final, debe estar funcional.
2. Defensa del proyecto 15%.
3. 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.
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, 2 y 3 se calculará la Nota Final del Proyecto.
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 el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
10.4. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
10.5. 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.
10.6. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de 0.
10.7. No presentarse a la revisión se obtiene nota de 0.

11. Cada grupo tendrá como máximo 60 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 (2018-10-04). Recuperado de: https://angularjs.io

Bootstrap Themes & Templates (2018-10-04). Recuperado de: https://wrapbootstrap.com/

How to Write Doc Comments for the Javadoc Tool. (2018-10-04). Recuperado de: http://www.oracle.com/technetwork/articles/java/index-137868.html

C# Coding Conventions (C# Programming Guide). (2018-10-04). Recuperado de: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions