
Tecnológico de Costa Rica

**Customer Relationship Manager
Especificación de diseño**

Version 1.2

Índice

| | |
|---|-----------|
| Información del documento | 2 |
| Descripción | 2 |
| Identificación del documento | 2 |
| Organización encargada | 2 |
| Historial de cambios | 2 |
| Introducción | 3 |
| Alcance | 3 |
| Referencias | 3 |
| Glosario | 3 |
| Metas y limitaciones arquitectónicas | 3 |
| Plataforma objetivo | 4 |
| Tecnologías a utilizar | 4 |
| Descripción y solución de los requerimientos | 4 |
| Restricciones globales | 4 |
| Suposiciones | 4 |
| Características de calidad a cumplir | 5 |
| Requerimientos | 5 |
| Seguimientos de ventas | 5 |
| Manejo de contactos | 6 |
| Portal de clientes | 7 |
| Reportes | 8 |
| Manejo de usuarios | 8 |
| Integración | 9 |
| Metodología de desarrollo | 10 |
| Planificación de la iteración | 10 |
| Ejecución de la iteración | 11 |
| Inspección y adaptación | 11 |
| Vista de implementación | 11 |
| Vista de datos | 12 |
| Estándar de código | 13 |
| Guía de diseño de la interfaz | 13 |

1. Información del documento

1.1. Descripción

El siguiente documento especifica el diseño que será la base del producto *Customer Relationship Manager*, referido de ahora en adelante como CRM. En este se definirán las características y la forma en cómo se van a llevar a cabo la implementación de los requerimientos del producto.

1.2. Identificación del documento

DS092017-Final

1.3. Organización encargada

Los siguientes recursos componen la organización encargada de llevar a cabo el diseño del sistema.

- Izcar Muñoz
- Isaac Campos
- Jeffrey Alvarado

1.4. Historial de cambios

| Fecha | Versión | Descripción de los cambios | Autor |
|------------|---------|---|------------------|
| 17/09/2017 | 1.0 | Primera versión | Jeffrey Alvarado |
| 21/09/2017 | 1.1 | Revisión de las metas y limitaciones arquitectónicas y de las guías de diseño. | Jeffrey Alvarado |
| 24/09/2017 | 1.2 | Inclusión de la solución a los requerimientos. | Jeffrey Alvarado |
| 19/10/2017 | 1.3 | Revisión del diseño del requerimiento 1. Revisión del modelo de bases de datos. | Jeffrey Alvarado |

2. Introducción

2.1. Alcance

El propósito de este documento es proporcionar la información y el marco requerido para implementar los requerimientos del producto de software CRM.

2.2. Referencias

| | |
|-------|---|
| [DES] | Documento de especificación de software |
| [DEP] | Documento estándar de programación |

2.3. Glosario

Las siguientes abreviaciones son usadas en el presente documento:

| | |
|-----|-----------------------------------|
| CRM | Customer Relationship Manager |
| API | Application programming interface |

3. Metas y limitaciones arquitectónicas

En esta sección se enumeran todos los requerimientos no funcionales relacionados con el rendimiento y la distribución que debe seguir el sistema, además de las características a las que tiene que ajustarse.

El sistema debe seguir el patrón arquitectónico Cliente-Servidor. Este debe cumplir con un nivel alto de cohesión y un nivel bajo de acoplamiento, donde cada módulo sea lo más independiente posible del resto del sistema y sea capaz de ser reemplazado y modificado afectando a la menor cantidad de funcionalidades del mismo. El sistema debe de ser web. Al mismo tiempo, debe cumplir con las siguientes características no funcionales:

- Ejecución con distintos tamaños de pantalla;
- Seguridad para evitar ataques informáticos;
- Alta tolerancia a fallos;
- Alta disponibilidad;
- Alta escalabilidad;

- Debe ser rápido.

3.1. Plataforma objetivo

El producto debe ser capaz de correr las siguientes características para el cliente:

- Windows 10, versión 1703;
- Google Chrome, versión 61;

El servidor debe cumplir con las siguientes características:

- Windows 10, versión 1703;
- MySQL, versión 5.7;
- IIS, versión 10;

3.2. Tecnologías a utilizar

A continuación se enlistan las herramientas que serán utilizadas para el desarrollo del producto:

- C# como lenguaje de desarrollo del lado servidor;
- Javascript como lenguaje de desarrollo del lado cliente;
- MySQL como motor de bases de datos;
- IIS como servidor del sitio web.

4. Descripción y solución de los requerimientos

4.1. Restricciones globales

1. Todas las funcionalidades del sistema, a excepción del portal de clientes, pueden realizarse únicamente después de que el usuario es autorizado por el sistema para hacerlas.

4.2. Suposiciones

1. El manejo de productos se lleva a cabo en una parte distinta del sistema especificado.
2. El subsistema obtiene la información de sus productos del módulo de mercadeo.

4.3. Características de calidad a cumplir

El sistema debe cumplir con las siguientes características de calidad:

- Portabilidad a distintos equipos como computadoras, teléfonos, tablets
- Seguridad para evitar ataques y hackeos
- Tolerante a fallos
- Escalabilidad
- Alta disponibilidad
- Buen rendimiento

4.4. Requerimientos

1. *Seguimientos de ventas*

1.1. Registro de productos vendidos;

1.1.1. Descripción

Se registran las ventas confirmadas de los productos.

1.1.2. Datos

- Nombre del producto
- Fecha de la venta
- Comisión del vendedor sobre el precio de venta
- Descuento sobre el precio de venta
- Precio final de la venta (calculado)
- Vendedor encargado de la venta (usuario)

1.1.3. Restricciones

- El descuento debe ser un número mayor o igual a 0
- La comisión de venta debe ser un número mayor o igual a 0
- Deben existir productos en el catálogo de ventas
- Depende de la integración con el sistema de mercadeo

1.2. Registro de las propuestas de venta;

1.2.1. Descripción

Manejo de todas las propuestas de ventas realizadas a clientes por parte de los vendedores (usuarios). Estas se envían como correo electrónico al cliente.

1.2.2. Datos

- Producto a vender
- Precio propuesto
- Cliente

1.2.3. Restricciones

- Deben existir productos en el catálogo de ventas
- Deben existir clientes como contactos

1.3. Registro de las respuestas a las propuestas de venta.

1.3.1. Descripción

Las respuestas a las propuestas de ventas mencionadas en el punto anterior deben ser registradas y asociadas a la misma propuesta.

1.3.2. Datos

- Estado de la propuesta (positiva o negativa).
- Descripción de la respuesta.
- Fecha.

1.3.3. Restricciones

No aplica

2. *Manejo de contactos*

2.1. Registro de contactos;

2.1.1. Descripción

Se registra la información de distintas entidades como contactos de clientes para ser accedidos por el usuario u otras partes del sistema.

Hay dos tipos de contacto: personas o empresas

2.1.2. Datos

- Nombre
- Apellidos (sólo personas)
- Dirección
- Teléfono
- Correo electrónico (sólo para personas)
- Empresa a la que pertenece (sólo personas, opcional)

2.1.3. Restricciones

No aplica

2.2. Visualización de contactos;

2.2.1. Descripción

Mostrar los datos de contactos registrados por el usuario

2.2.2. Datos

No aplica

2.2.3. Restricciones

- Deben existir contactos registrados

3. *Portal de clientes*

3.1. Información de puntos de contacto de la empresa;

3.1.1. Descripción

Mostrar todos los puntos de contacto con los que cuenta la empresa

3.1.2. Datos

No aplica

3.1.3. Restricciones

- Deben existir puntos de contacto registrados en el sistema.

3.2. Productos relacionados

3.2.1. Descripción

Mostrar información de los productos de la empresa.

3.2.2. Datos

No aplica

3.2.3. Restricciones

- Deben existir productos registrados en el sistema.
- Depende de la integración con el sistema de mercadeo.

3.3. Registro de errores en productos;

3.3.1. Descripción

Registro de reportes de errores de los productos vendidos a un cliente.

3.3.2. Datos

- Correo electrónico del denunciante
- Producto
- Fecha del error
- Descripción del error

3.3.3. Restricciones

- Deben existir productos registrados en el sistema.

- Depende de la integración con el sistema de mercadeo.

3.4. Chat.

3.4.1. Descripción

Chat en tiempo real con algún encargado dentro de la empresa

3.4.2. Datos

- Mensaje del cliente

3.4.3. Restricciones

No aplica

4. *Reportes*

4.1. Generar reportes del seguimiento de ventas

4.1.1. Descripción

Generar un reporte a partir de los datos de ventas según un intervalo específico de tiempo.

4.1.2. Datos

- Intervalo de fechas (opcional)
- Productos (opcional)

4.1.3. Restricciones

- Deben haber ventas registradas.
- Deben haber ventas registradas en el período especificado.

4.2. Generar reportes de errores de productos.

4.2.1. Descripción

Generar un reporte a partir de los reportes de errores de los clientes según un intervalo específico de tiempo.

4.2.2. Datos

- Intervalo de fechas (opcional)
- Productos (opcional)

4.2.3. Restricciones

- Deben haber reportes de errores registrados.
- Deben haber reportes de errores registrados en el intervalo de tiempo especificado

5. *Manejo de usuarios*

5.1. Registro de usuarios;

5.1.1. Descripción

Registros de usuarios nuevos en el sistema.

5.1.2. Datos

- Correo electrónico
- Contraseña

5.1.3. Restricciones

- La contraseña debe ser de al menos 6 caracteres.
- El correo electrónico debe tener un formato válido.
- El correo electrónico no debe estar registrado en la base de datos.

5.2. Ingreso de usuarios.

5.2.1. Descripción

Autorización de usuarios existentes para el uso del sistema

5.2.2. Datos

- Correo electrónico
- Contraseña

5.2.3. Restricciones

No aplica.

6. Integración

6.1. Integración con la plataforma de correos;

6.1.1. Descripción

Capacidad del sistema de enviar correos electrónicos mediante la plataforma encargada de esta función.

6.1.2. Datos

No aplica

6.1.3. Restricciones

No aplica

6.2. Integración con la IP de los teléfonos;

6.2.1. Descripción

Integración del sistema VoIP de la empresa con el sistema de contactos.

6.2.2. Datos

No aplica

6.2.3. Restricciones

Debe de haber un programa de VoIP funcionando en la empresa.

6.3. Integración con el módulo de mercadeo.

6.3.1. Descripción

Capacidad de comunicarse con el módulo de mercadeo, encargado campañas publicitarias y manejo de productos.

6.3.2. Datos

No aplica

6.3.3. Restricciones

No aplica

5. Metodología de desarrollo

El proyecto se ejecuta en bloques cortos y fijos, estas iteraciones tienen un plazo de duración que va desde una semana hasta un mes. La función de cada iteración es obtener como resultado un incremento del producto final que sea entregado al cliente cuando se le solicite. El proceso parte de la lista de requisitos del producto, o *User Stories*, el cliente prioriza los objetivos y quedan repartido en iteraciones y entregas. Las actividades que se llevan son las siguientes:

Planificación de la iteración

El primer día de la iteración se realiza una reunión de planificación de iteración que se divide en dos partes:

Selección de requisitos: El cliente presenta al equipo la lista de requisitos del producto. El equipo pregunta al cliente las dudas y selecciona los requisitos más prioritarios.

Planificación de la iteración: El equipo desarrolla la lista de actividades de la iteración para desarrollar los requisitos evaluados por el cliente.

Ejecución de la iteración

Cada día el equipo realiza una reunión corta o *Daily Scrum*, esto con el fin de inspeccionar el el trabajo que el resto está realizando para poder las correcciones para lograr el objetivo. En la reunión se responden tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el *Scrum Master* se encarga de que el equipo:

- Cumplir con su compromiso
- Eliminar obstáculos que el equipo no puede resolver.
- Proteger al equipo de interrupciones externas que puedan afectarlos, de ahí el nombre de la metodología.
- El cliente junto con el equipo actualiza los requerimientos, si es necesario cambian los objetivos del proyecto.

Inspección y adaptación

El último día de la iteración se realiza la reunión, tiene dos partes:

Demostración: El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado.

Retrospectiva del equipo: El equipo se autoanaliza su manera de trabajo y cuáles son los problemas de su futuro impedimento.

6. Vista de implementación

La arquitectura a seguir es Cliente-Servidor. La arquitectura Cliente-Servidor describe la relación de los programas que cooperan en una aplicación. El componente de servidor proporciona una función o servicio a uno o varios clientes, los cuales inician solicitudes de dichos servicios. Ambos componentes operan independientemente uno del otro, y pueden operar en distintos equipos o en el mismo.

```
graph LR
    Actor[Actor] -.-> Navegador[Navegador web]
    Navegador -.-> Servidor
    subgraph Servidor
        ServidorWeb[Servidor web]
        Conexion[«interface» Conexión de base de datos]
        ServidorWeb -.-> Conexion
    end
    subgraph MotorBaseDatos[Motor de base de datos]
        BaseDatos[Base de datos]
    end
    BaseDatos -.-> Conexion
```

El modelo de base de datos es relacional. Este modelo organiza los datos en una o más tablas (o "relaciones") de columnas y filas, con una clave única que identifica cada fila. Las filas también se llaman registros o tuplas. Las columnas también se llaman atributos. Generalmente, cada tabla / relación representa un "tipo de entidad" (como cliente o producto). Las filas representan instancias de ese tipo de entidad y las columnas que representan los valores atribuidos a esa instancia.

Diagrama de classes i relacions per a un sistema de venda online. Les classes són:

- empresa**: id empresa, descrició empresa, id int d'admissió
- resposta**: id resposta, id prequest, id producte, id correu, id venda, id usuari, id persona, id telefon, data
- prequest**: id prequest, id producte, id correu, id venda, id usuari, id persona, id telefon, data
- producte**: id producte, nom, preu
- correu**: id correu, descrició correu, id usuari, id persona, id telefon
- venda**: id venda, id producte, id correu, id usuari, id persona, id telefon, data
- usuari**: id usuari, descrició usuari, id correu, id venda, id persona, id telefon
- persona**: id persona, nom, id prequest, id producte, id correu, id venda, id telefon
- telefon**: id telefon, id usuari, id persona, id venda

Relacions:

- empresa** a **resposta**: 1 a moltes
- empresa** a **prequest**: 1 a moltes
- empresa** a **producte**: 1 a moltes
- empresa** a **correu**: 1 a moltes
- empresa** a **venda**: 1 a moltes
- empresa** a **usuari**: 1 a moltes
- empresa** a **persona**: 1 a moltes
- empresa** a **telefon**: 1 a moltes
- resposta** a **prequest**: 1 a moltes
- resposta** a **producte**: 1 a moltes
- resposta** a **correu**: 1 a moltes
- resposta** a **venda**: 1 a moltes
- resposta** a **usuari**: 1 a moltes
- resposta** a **persona**: 1 a moltes
- resposta** a **telefon**: 1 a moltes
- prequest** a **producte**: 1 a moltes
- prequest** a **correu**: 1 a moltes
- prequest** a **venda**: 1 a moltes
- prequest** a **usuari**: 1 a moltes
- prequest** a **persona**: 1 a moltes
- prequest** a **telefon**: 1 a moltes
- producte** a **correu**: 1 a moltes
- producte** a **venda**: 1 a moltes
- producte** a **usuari**: 1 a moltes
- producte** a **persona**: 1 a moltes
- producte** a **telefon**: 1 a moltes
- correu** a **venda**: 1 a moltes
- correu** a **usuari**: 1 a moltes
- correu** a **persona**: 1 a moltes
- correu** a **telefon**: 1 a moltes
- venda** a **usuari**: 1 a moltes
- venda** a **persona**: 1 a moltes
- venda** a **telefon**: 1 a moltes
- usuari** a **persona**: 1 a moltes
- usuari** a **telefon**: 1 a moltes
- persona** a **telefon**: 1 a moltes

8. Estándar de código

El sistema debe de seguir el estándar de código definido en el documento [DEP]. En este se especifica a detalle las convenciones y recomendaciones de la estructura de código del sistema.

9. Guía de diseño de la interfaz

El diseño del sistema se basa en las tendencias del mercado y los requerimientos del cliente, con el fin de satisfacer sus necesidades, se utilizará Bootstrap como framework CSS, el cual se basa en un *grid* de 12 columnas, además se hará uso de bibliotecas como:

- **metisMenu:** Es un plugin de jQuery el cual nos permite crear menús con una excelente calidad gráfica y con mucha flexibilidad.
- **jQuery-ui** como un auxiliar al framework para JavaScript jQuery, la cual nos permite utilizar componentes como:
 - **Accordeon:** Menú con efecto acordeón.
 - **Autocomplete:** Caja con autocompletado.
 - **Button:** Botón.
 - **Dialog:** Ventanas con contenido.
 - **Slider:** Elemento para elegir en un rango de valores.
 - **Tabs:** Pestañas.
 - **Datepicker:** Calendario gráfico.
 - **Progressbar:** Barra de progreso.

Además nos permite crear interacciones con los diferentes elementos de la página, por medio de Una API para añadir transiciones animadas y facilidades para interacciones.

- **Peity** y **Sparklines** y **Flot** para la creación de los gráficos se utilizará algunos plugins de jQuery como Peity, Sparklines o Flot, con los cual podremos crear distintos tipo de gráficos como: mini pie, tipo dona, gráfico de lines o gráficos de barra. Esto nos garantiza una compatibilidad con cualquier navegador que sea compatible con el formato SVG

Para la iconografía del sitio se utilizará la biblioteca de CSS llamada *Font Awesome* la cual nos permite utilizar una gran cantidad de iconos con un excelente nivel de detalle y calidad.