Proyecto Final - CRM API

**UNIVERSIDAD TÉCNICA NACIONAL**

**CARRERA: INGENIERÍA DEL SOFTWARE**

**CURSO: PROGRAMACION EN AMBIENTE WEB II**

**CÓDIGO: ISW-711**

**VALOR: 100 puntos, 30%**

**PROFESOR: Bladimir Arroyo**

## Objetivo

Desarrollar una aplicación basada en Arquitecturas Orientadas a Servicios utilizando REST como estilo de arquitectura.

## Requerimientos

### Funcionales

Usando la estructura típica de un mantenimiento desarrolle los siguientes mantenimientos con sus condiciones específicas, el mantenimiento consiste en Listar, Crear, Actualizar, Ver y Eliminar cada uno de los distintos componentes descritos más adelante:

1. Listar: será la pantalla inicial de cada módulo, mostrará un listado de todos los registros asociados.
2. Crear: permitirá crear un nuevo registro
3. Actualizar: el usuario podrá ver un registro en detalle y luego actualizarlo
4. Ver: permitiría ver todo el detalle de un registro
5. Eliminar

Módulos a Desarrollar:

1. Autenticación
   1. Los usuarios podrán ingresar con usuario y clave.
   2. Solo el usuario administrador puede administrar los usuarios. Crear, Ver, Editar y Eliminar.
   3. Una vez autenticado se genera un token (pueden usar JWT) y en cada request se debe validar que el token sea válido y que identifique al usuario autenticado de manera que se pueda validar si cuenta con los permisos para realizar la acción.

|  |
| --- |
| Session |
| ID |
| USERNAME |
| PASSWORD |
| TOKEN |
| PERMISOS |

1. Usuarios, el administrador será capaz de administrar los usuarios del sistema, los usuarios tendrán la siguiente información:
   1. Nombre
   2. Apellidos
   3. Username
   4. Contraseña

|  |
| --- |
| User |
| ID |
| NAME |
| LASTNAME |
| USERNAME |
| PASSWORD |

1. Clientes, los clientes se refiere a las empresas clientes, todos los usuarios van a tener la capacidad de Administrar clientes, los clientes estarán compuestos por la siguiente información:
   1. Nombre
   2. Cédula Jurídica
   3. Página Web
   4. Dirección Física
   5. Número de Telefono
   6. **Sector ( Dropdown con los siguientes valores: Educación, Industria, Agricultura, Manufactura, Servicios, Otros)**

|  |
| --- |
| Client |
| ID |
| NAME |
| IDENTITY\_CARD |
| WEB\_PAGE |
| ADDRESS |
| PHONE |
| **SECTOR** |

1. Contactos, los contactos son las personas asociadas a un Cliente, todos los usuarios podrán crear contactos a un Cliente. Los contactos estarán compuestos por la siguiente información:
   1. Cliente al que pertenece
   2. Nombre
   3. Apellidos
   4. Correo electrónico
   5. Número telefónico
   6. Puesto

|  |
| --- |
| Contacts |
| ID |
| CLIENT\_ID |
| NAME |
| LAST\_NAME |
| EMAIL |
| PHONE |
| JOB |

1. Reuniones, los usuarios pueden administrar reuniones, las reuniones estarán compuestas por la siguiente información
   1. Título de la reunión
   2. Dia y hora
   3. Usuario asignado (Listado de Usuarios del sistema, debe seleccionar al menos uno)
   4. Es Virtual (checkbox)
   5. Cliente (opcional)

|  |
| --- |
| Meeting |
| ID |
| TITLE |
| DAY\_TIME |
| USER\_ID |
| IS\_VIRTUAL |
| CLIENT\_ID |
|  |

1. Support Tickets, son solicitudes de soporte de parte del cliente, estarán compuestas por:
   1. Título del Problema
   2. Detalle del problema
   3. Quién reportó el problema
   4. Cliente
   5. Estado actual (Abierto, En Proceso, En Espera, Finalizado)

### Técnicos

Con respecto a los requerimientos técnicos del proyecto se espera que:

* Desarrollar tanto el cliente como el servicio Web API.
* El Web API deberá usar el estilo de arquitectura REST con sus restricciones y características.
* Puede utilizar cualquier lenguaje de programación o framework tanto para el cliente (consumidor) como para el servicio.
* Deberán realizar las validaciones de datos tanto en el cliente como en el Web API.
* Desarrollar una documentación del Web API usando https://swaggerhub.com

## Aspectos Administrativos:

* El proyecto se realizará de manera individual
* La fecha de entrega del proyecto será el 4 de Mayo del 2018.

**Evaluación**

El proyecto será evaluado de acuerdo a la siguiente tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requerimiento** | **Puntaje** | | | |
| Autenticación y autorización | **0 pts** | **5 pt** | **8 pts** | **15 pts** |
| No se desarrolló del todo | Desarrollo el API pero tiene errores. | Desarrolló el API y funciona pero tiene errores al consumirlo con el cliente. | Desarrollo tanto el API como el Cliente y funcionan sin errores. |
| CRUD de Usuarios | **0 pts** | **5 pt** | **8 pts** | **10 pts** |
| No se desarrolló del todo | Desarrollo el API pero tiene errores. | Desarrolló el API y funciona pero tiene errores al consumirlo con el cliente. | Desarrollo tanto el API como el Cliente y funcionan sin errores. |
| CRUD de Clientes | **0 pts** | **5 pt** | **8 pts** | **10 pts** |
| No se desarrolló del todo | Desarrollo el API pero tiene errores. | Desarrolló el API y funciona pero tiene errores al consumirlo con el cliente. | Desarrollo tanto el API como el Cliente y funcionan sin errores. |
| CRUD de Contactos | **0 pts** | **5 pt** | **8 pts** | **10 pts** |
| No se desarrolló del todo | Desarrollo el API pero tiene errores. | Desarrolló el API y funciona pero tiene errores al consumirlo con el cliente. | Desarrollo tanto el API como el Cliente y funcionan sin errores. |
| CRUD de Reuniones | **0 pts** | **5 pt** | **8 pts** | **10 pts** |
| No se desarrolló del todo | Desarrollo el API pero tiene errores. | Desarrolló el API y funciona pero tiene errores al consumirlo con el cliente. | Desarrollo tanto el API como el Cliente y funcionan sin errores. |
| CRUD de Tiquetes de Soporte | **0 pts** | **5 pt** | **8 pts** | **10 pts** |
| No se desarrolló del todo | Desarrollo el API pero tiene errores. | Desarrolló el API y funciona pero tiene errores al consumirlo con el cliente. | Desarrollo tanto el API como el Cliente y funcionan sin errores. |
| Buenas Prácticas: Identación, convenciones y documentación de código | **0 pts** | **2 pt** | **3 pts** | **5 pts** |
| Código desordenado, no cumple convenciones de código ni documentación | Existen Convenciones de Código, estilo y documentación mínima | Existen convenciones de Código, estilo y documentación básica | Existe en buen detalle con respecto convenciones, estilo y documentación de código. |
| Defensa del Proyecto | **0 pts** | **5 pt** | **10 pts** | **20 pts** |
| No supo responder correctamente las preguntas realizadas por el profesor. | Respondió con deficiencia las preguntas realizadas por el profesor. | Respondió correctamente las preguntas realizadas por el profesor, pero no se nota seguridad al responder las preguntas. | Respondió correctamente las preguntas realizadas por el profesor y se nota dominio del tema y completa autoría del proyecto. |
| Patrón REST | **0 pts** | **2 pt** | **3 pts** | **5 pts** |
| Posee más de 3 errores en la implementación del patrón REST | Posee más de 2 errores en la implementación del patrón REST | Posee 1 error en la implementación del patrón REST | Realizó una correcta implementación del patrón REST |
| Documentación | **0 pts** | **2 pt** | **3 pts** | **5 pts** |
| No realizó documentación el Web API | Realizó la documentación de algunos de los APIs únicamente. | Realizó la documentación de la mayoría de los APIs | Realizó la documentación completa de todos los APIs |