

Proyecto Final - CRM API

UNIVERSIDAD TÉCNICA NACIONAL
CARRERA: INGENIERÍA DEL SOFTWARE

CURSO: PROGRAMACIÓN 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

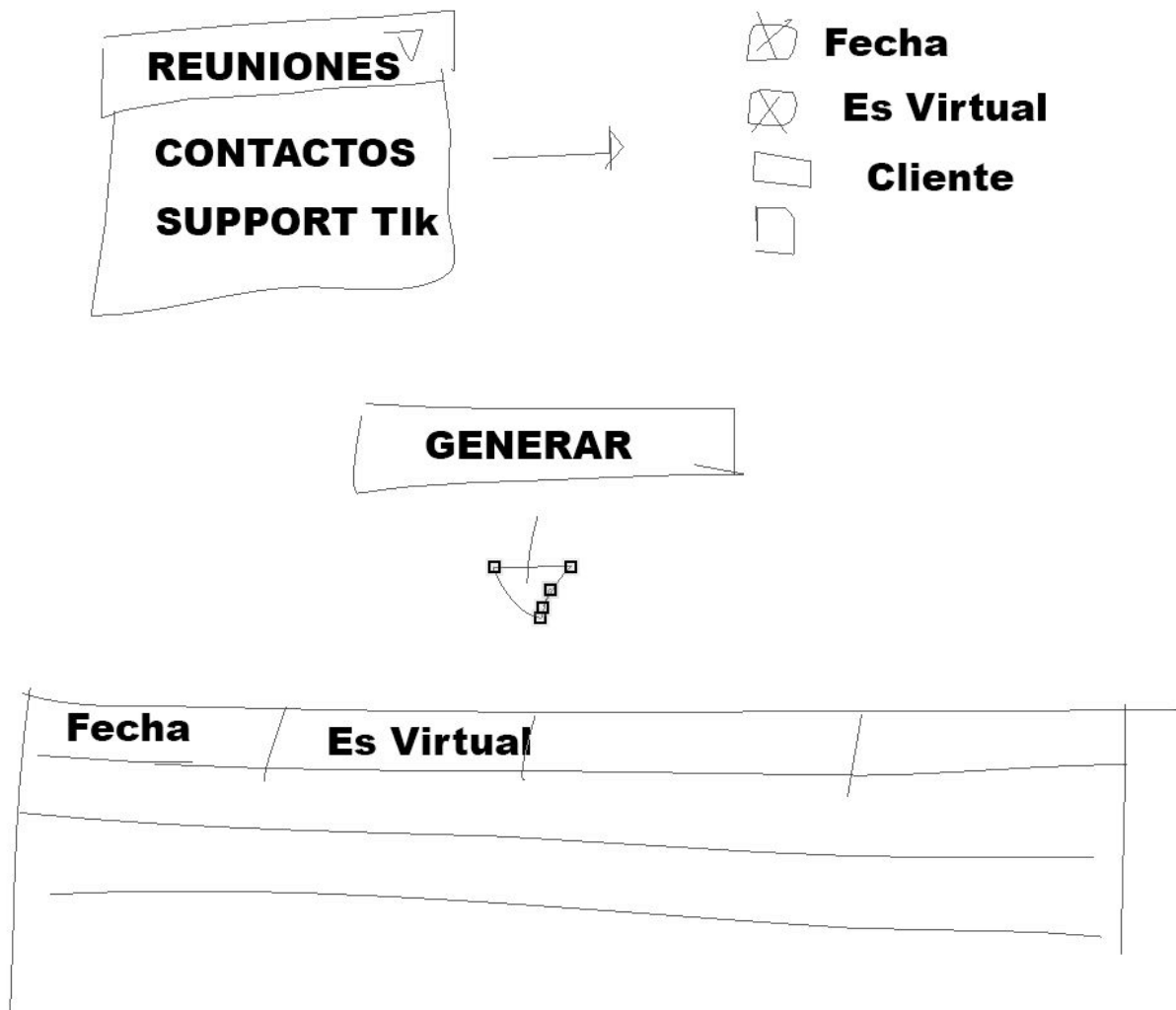
Módulos a Desarrollar:

1. Autenticación
 - a. Los usuarios podrán ingresar con usuario y clave.
 - b. Solo el usuario administrador puede administrar los usuarios. Crear, Ver, Editar y Eliminar.
 - c. 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.
2. Usuarios, el administrador será capaz de administrar los usuarios del sistema, los usuarios tendrán la siguiente información:

- a. Nombre
 - b. Apellidos
 - c. Username
 - d. Contraseña
- 3. 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:
 - a. Nombre
 - b. Cédula Jurídica
 - c. Página Web
 - d. Dirección Física
 - e. Número de Telefono
 - f. Sector (Dropdown con los siguientes valores: Educación, Industria, Agricultura, Manufactura, Servicios, Otros)
- 4. 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:
 - a. Cliente al que pertenece
 - b. Nombre
 - c. Apellidos
 - d. Correo electrónico
 - e. Número telefónico
 - f. Puesto
- 5. Reuniones, los usuarios pueden administrar reuniones, las reuniones estarán compuestas por la siguiente información
 - a. Título de la reunión
 - b. Dia y hora
 - c. Usuario asignado (Listado de Usuarios del sistema, debe seleccionar al menos uno)
 - d. Es Virtual (checkbox)
 - e. Cliente (opcional)
- 6. Support Tickets, son solicitudes de soporte departe del cliente, estarán compuestas por:
 - a. Título del Problema
 - b. Detalle del problema
 - c. Quién reportó el problema
 - d. Cliente
 - e. Estado actual (Abierto, En Proceso, En Espera, Finalizado)
- 7. Extras: Reportes con GraphQL.
 - a. En la interfaz sugerida el usuario podrá seleccionar la entidad de la que desea realizar un reporte (Reuniones, Contactos, Clientes, Support Tickets). Una vez seleccionada la entidad se le mostrará al usuario los posibles atributos que se pueden incluir en el reporte sobre esa entidad (cada uno de los campos disponibles en la base de datos), el usuario podrá seleccionar uno o más de

esos atributos y al presionar el botón “Generar” el sistema mostrará un reporte con los datos que correspondan (entidad y atributos seleccionados unicamente).

b. Bosquejo:



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> o similar

Aspectos Administrativos:

- El proyecto se realizará de manera individual
- La fecha de entrega del proyecto será el 28 de Agosto del 2020.

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	Desarrollo tanto el API como el Cliente y

			consumirlo con el cliente.	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
Extras				
Reportes con GraphQL	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.