



Proyecto de Ingeniería de Software 2

Examen Técnico I

Profesor

Dennis Córdoba López

Estudiante

Milton Pineda Lizardo

Fecha de entrega

25 de junio de 2019

II cuatrimestre 2019

PREGUNTAS SOBRE APIs

a. ¿Qué es Twilio?

Twilio es una plataforma que permite comunicaciones en la nube. El servicio la ofrece la empresa la cual tiene una sede en San Francisco California. En mayor detalle Twilio le permite a desarrolladores de software recibir llamadas de teléfono, enviar y recibir mensajes SMS por medio de la programación y además realizar otras funciones de comunicación por medio de APIs de servicios web.

b. ¿Cuál es el nombre del NuGet que permite enviar SMS que se va a utilizar en el curso? Twilio.

El nombre del NuGet es Twilio.

c. Describa los pasos necesarios para poder hacer uso del API de Paypal.

- Ingresar a la plataforma de PayPal y registrar una cuenta.
- Acceder a la sección de "Accounts".
- Seleccionar una cuenta y luego "actualizar contraseña/update password".
- Iniciar sesión con credenciales respectivos.
- Entrar a sandbox.paypal.com.
- Seleccionar tools>all tools.
- Seleccionar Paypal buttons;
- Crear button:
 - Seleccionar "Simple buy now button".
 - Llenar la información de tipo de moneda, tarifa y cantidad de items.
 - Seleccionar "Crear button".
 - Incorporar html proveído al proyecto deseado.
- Ejecutar proyecto.
- Presionar el botón previamente creado.
- Llenar formulario.
- Selecciona "comprar"
 - Ingresar con cuenta de comprador
 - Utilizar tarjeta asociada
 - Ver factura

d. ¿Qué servicios provee el servicio de Cloudinary?

Permite la administración de documentos, videos e imágenes en la nube, se puede almacenar, manipular y subir las imágenes y videos a aplicaciones web.

e. Explique brevemente lo que es un "SaaS"

Software as a service.

“Un método de entrega y licencia de software en el que se accede al software en línea a través de una suscripción, en lugar de comprarlo e instalarlo en computadoras individuales.” -Google dictionary.

Se entiende que los usuarios pueden acceder al un software por medio del pago de una licencia/suscripción. El acceso al software es por medio de conexión de internet.

f. ¿Por qué el formato de los códigos QR es cuadrado?

Su formato viene de su capacidad de codificar caracteres kanji y de la utilidad de expresar datos en direcciones, que aparte de crear una lectura rápida ocupa menos espacio que su antecesor, el código de barras.

g. ¿Qué es la “quite zone” y por qué es importante?

Corresponde al espacio mínimo de cuatro módulos. Entendiendo módulo como el cuadrado mínimo presente en cada código QR, que se debe dejar de espacio para que pueda ser leído y ubicado por el lector.



h. ¿Qué es SendGrid?

“SendGrid es una plataforma de comunicación para clientes basada en Denver, Colorado, para correo electrónico transaccional y de marketing.” - SendGrid.com

“SendGrid proporciona un servicio de entrega de correo electrónico basado en la nube que ayuda a las empresas con la entrega de correo electrónico. El servicio gestiona varios tipos de correo electrónico, incluidas notificaciones de envío, solicitudes de amistad, confirmaciones de registro y boletines informativos por correo electrónico”

Se entiende que sendGrid es una empresa la cual brinda servicios de entrega de correo electrónico por medio de la nube.

i. ¿Qué es un nuget y como se instala?

SendGrid nuget

Se puede instalar en un proyecto de visual studio al acceder a Nuget Package Management y buscar el nombre del nuget. Seguidamente se hace click en install.

j. ¿Qué es un action callback en Telegram y para qué se utiliza?

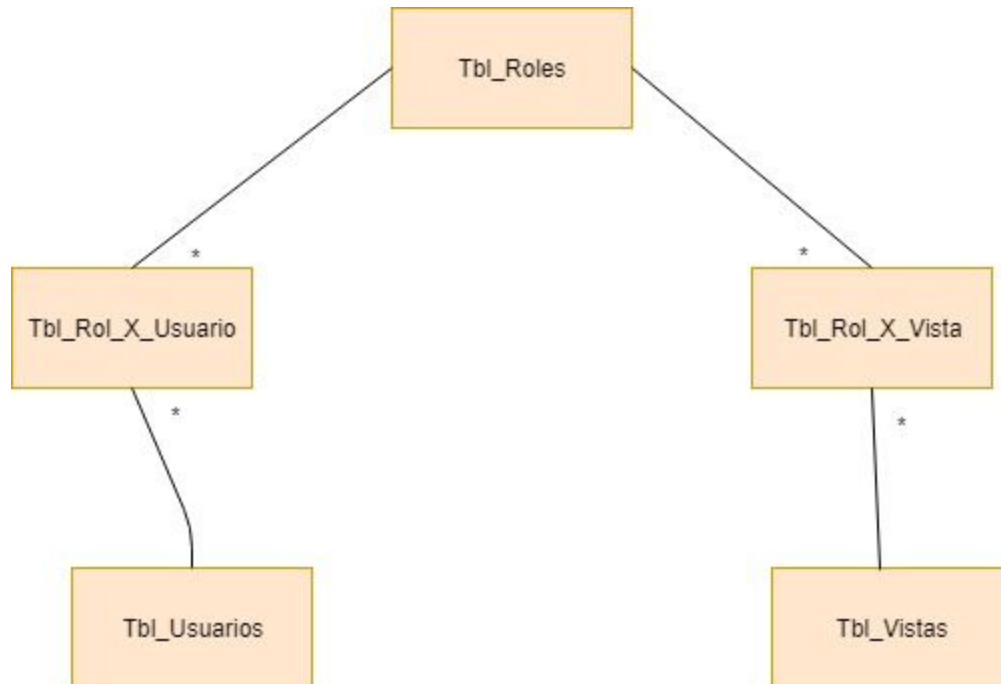
“Simply put: A callback is a function that is to be executed after another function has finished executing — hence the name ‘call back’.” -Codeburst.io

Se puede decir que se usa para realizar una acción dependiendo de lo que el usuario ingrese. Es decir se le puede solicitar información al usuario, esperar a que este responda y dependiendo de la respuesta ejecutar una acción.

Modelo de datos.

a. Explique el modelo de seguridad que debe utilizar el proyecto. (roles, usuarios, vistas), incorpore un diagrama en su explicación.

Se toma en consideración tres tipos de entidades en cuanto a la seguridad: roles, usuarios y vistas. La asociación entre estas permitirá a da distintos usuarios acceder a las mismas vistas y dentro de estar poder realizar distintas acciones dependiendo del rol que estos tengan.



A Partir del diagrama anterior podemos apreciar que la tabla “Tbl_Rol_Usuario” contiene la información de los usuarios y los roles respectivos este pueda tener. Junto con lo anterior en la tabla “Tbl_RolX_Vista” contiene la información de las vistas y cuáles roles tienen acceso a estas. Si se utiliza el método anterior se puede efectuar el siguiente escenario:

- El usuario “Juan” tiene el rol de “Administrador” (Tbl_Rol_Usuario)
- El usuario “Roberto” tiene el rol de “Administrador” (Tbl_Rol_Usuario)
- El usuario “David” tiene el rol de “Empleado” (Tbl_Rol_Usuario)
- La vista “Bitacora” tiene el rol “Administrador”. (Tbl_RolX_Vista)

Con los datos anteriores podríamos decir que existe una vista llamada “Bitacora” a la que solo pueden acceder Juan y Roberto ya que poseen el rol de “Administrador” el cual está asociado a la vista.

Arquitectura

a. Detalle con sus propias palabras, el objetivo de las siguientes clases:

i. SqlDao: Clase encargada de realizar la conexión con la base de datos. Recibe la operacionSQL por parte del CrudFactory. Le da formato a los parametros que trae la operacionSQL.

ii. Mapper: El mapper es la clase utilizada por el CrudFactory respectivo de cada entidad para obtener la operación SQL (junto con los parámetros a enviar y los procedimientos almacenados) para hacer una petición a la base de datos. Es decir, recibe un BaseEntity del CrudFactory y lo convierte en algo que la base de datos pueda entender.

iii. Manager: Recibe peticiones del controller (el cual está ubicado en el Web API). Este hace llamados al crud factory respectivo de cada entidad. Permite manejar validaciones de datos antes de acceder a lo que es el DataAccess.

b. Explique la capa de manejo de excepciones, utilice los recursos que considere necesarios (Diagramas, videos, imágenes)

La capa manejo de excepciones nos permite trabajar con las excepciones controlables por el programador. Con el fin de ejemplificar podemos poner el caso hipotético de una aplicación para un casino. En esta aplicación se desea que todos los usuario que se intenten de registrar sean mayores de edad, por lo que si un usuario se registra con la fecha de nacimiento de un menor se puede detectar esto y por medio de la capa de datos activar la excepción 2 que en este caso hipotético corresponde al mensaje de la excepción de "Usuario menor de edad". El mensaje anterior está escrito de una manera que es más entendible para el usuario de la aplicación, de esta forma podrá tomar una acción correctiva sin necesidad de conocimiento sobre términos técnicos.

¿Cómo funciona?

Dentro de nuestra solución creamos un proyecto tipo class library y en este creamos una Clase BusinessException y además un Exception Manager. Al nosotros estar en la clase clienteCasinoManager (para mantener el ejemplo anterior) como programadores podemos dejar una condición que si el usuario al intentar registrarse dentro del método create es menor de edad tiramos un BusinessException y le enviamos como parámetro el índice que nosotros ya sabemos corresponde a la

excepción de menor de edad. Luego al tirar ese BusinessException (throws) cuando lleguemos al catch éste no va redirigir al Exception Manager donde el constructor nos hará entrar en un método llamado “ LoadMessages();” el cual nos permitirá acceder a Data access para así obtener el mensaje respectivo para la excepción dada.

```
        if (customer.Age >= 18)
            crudCustomer.Create(customer);
        else
            throw new BusinessException(2);
    }
    catch(Exception ex)
    {
        ExceptionManager.GetInstance().Process(ex);
    }
```

- CustomerManager

```
private static ExceptionManager instance;

private static Dictionary<int, ApplicationMessage> messages = new Dictionary<int, ApplicationMessage>();

private ExceptionManager()
{
    LoadMessages();
}
```

- Exception Manager

```
private void LoadMessages()
{
    var crudMessages = new AppMessagesCrudFactory();

    var lstMessages = crudMessages.RetrieveAll<ApplicationMessage>();

    foreach (var appMessage in lstMessages)
    {
        messages.Add(appMessage.Id, appMessage);
    }
}
```

-Exception manager > LoadMessages

	Id	Text
	0	Houston we ha...
	1	Invalid custom...
▶	2	er must be over 18
	3	This customer ...
	4	Sex must be 'm'...
*	<i>NULL</i>	<i>NULL</i>

-Tabla dentro de SQLSERVER