

Evaluación .NET

Generalidades

- **Fecha:** 09/04/2019
 - **Autor:** Alejandro Techera
 - **Contacto:** alejandro.techera@xpantion.com
-

Descripción de la Evaluación

Introducción

La prueba consiste en resolver una problemática dada utilizando como plataforma de base al Framework .NET (o .NET Core). Se solicita se utilice una arquitectura lo más desacoplada posible, separando las capas de **Presentación, Servicios/Negocio y Acceso a datos**.

Se lista una serie de guías y herramientas recomendadas en la sección de **Guías y Herramientas**, aunque el uso de las mismas no es mandatorio.

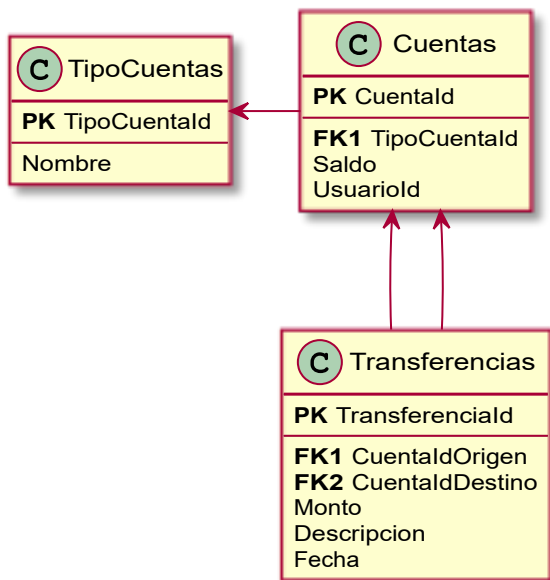
El esfuerzo estimado para la generación de los entregables es de **2 días**, por lo que es importante priorizar los principales aspectos de la solución y dejar documentados todos aquellos que no pudieron ser resueltos por falta de tiempo.

Problemática a resolver

Implementar la transferencia de fondos de un usuario registrado en un sistema de eBanking. Donde deberán tenerse en cuenta los siguientes elementos:

- La transferencia se hará entre cuentas del usuario.
- Deberá solicitarse: cuenta origen, cuenta destino, monto a transferir y una descripción para la transferencia.
- Deberán actualizarse los saldos de las cuentas involucradas al momento de realizar la transferencia.
- Se deberá validar que la cuenta origen tenga saldo suficiente para realizar la transferencia.
- El sistema eBanking mencionado, es un sistema de gran porte que contiene varios módulos entre ellos el módulo de Transferencias al cual hace referencia este caso.

El modelo de datos propuesto es el siguiente:



Dónde la relación con la tabla de usuarios se deja abierta a que se defina de la forma más conveniente para la implementación.

Entregables y Plazo

Entregables

Se deberán entregar los siguientes elementos:

- Solución de VisualStudio con el código C#.
- **Script** o **Backup** de la base de datos (**SQL Server**, indicar versión utilizada).
- Lista de **elementos no implementados** por motivos de tiempo, describiendo la forma en que se hubieran implementado.
- Lista de **supuestos y consideraciones** tomados en cuenta para la definición e implementación de la solución.
- **Documentos y diagramas** generados resultantes de las tareas de análisis y diseño (MER, diagramas de clases, especificación de requerimientos, etc.)

Este elemento NO es mandatorio.

- Consumo de **horas discriminado por tarea**:

Tarea	Horas
Análisis	xxx hs
Diseño técnico	xxx hs
Diseño/Ajustes estéticos	xxx hs
Desarrollo	xxx hs
QA/Testing	xxx hs

Plazo

El plazo estimado para la realización de la prueba es de **2 días**.

Aspectos que no se logren cubrir dentro del plazo podrán ser descritos como parte del entregable llegado el plazo, como se mencionó previamente.

Guías y Herramientas

Guías

Se listan a continuación guías que pueden tomarse como referencia:

Presentación

- MVC .NET: <https://dotnet.microsoft.com/apps/aspnet/mvc>
- Web Forms: <https://dotnet.microsoft.com/apps/aspnet/web-forms>
- Angular: <https://angular.io/>

Servicios

- WCF: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/getting-started-tutorial>
- .NET WebApi: <https://dotnet.microsoft.com/apps/aspnet/apis>

Acceso a datos

- EntityFramework: <https://docs.microsoft.com/en-us/ef/#pivot=ef6x>
- LINQ to SQL: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/linq/>
- Dapper: <https://dapper-tutorial.net/dapper>

Otros

- Guía de arquitectura: <https://archive.codeplex.com/?p=apparchguide>
- Package Manager (.NET): <https://www.nuget.org/>
- Package Manager (Node): <https://docs.npmjs.com/about-npm/>

Herramientas

Se listan a continuación las herramientas que podrán ser utilizadas. Esta lista no pretende limitar al uso de las mismas, sino servir como guía.

- Visual Studio Community: <https://visualstudio.microsoft.com/vs/community/>
- SQL Server 2017 - Developer: <https://go.microsoft.com/fwlink/?linkid=853016>
- SQL Server Management Studio: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>