Taller de Programación 2018
Trabajo Práctico Final

Introducción

El trabajo práctico final consta de la realización de una aplicación para una terminal autoservicio o kiosco bancario.

Una terminal de autoservicio o kiosco es un equipo informático situado en lugar público que permite a los usuarios realizar múltiples acciones. También se utiliza como herramienta de información y marketing para las empresas. En la actualidad, los kioscos interactivos a menudo tienen pantallas táctiles. Tienen como objetivo presentar una interfaz amistosa y de fácil interacción que facilite su utilización por cualquier tipo de usuario. Se utilizan en muchas aplicaciones y mercados verticales, tales como bancos, ventas de entradas de espectáculos, correos, hospitales, aeropuertos, estaciones ferroviarias, grandes supermercados. etcétera.

La aplicación deberá ser capaz de identificar al cliente, presentándole diferentes opciones de operaciones a realizar una vez identificado. La aplicación interactuará con una API de servicios del core bancario de la institución, con el fin de delegar las ejecuciones de las operaciones seleccionadas. Básicamente la aplicación es una orquestador de acceso a servicios.

Requerimientos funcionales

A continuación se describen los requerimientos funcionales necesarios que la aplicación debe satisfacer.

Identificación del cliente
La primera pantalla en presentarse al usuario es la de identificación de cliente. En la misma el usuario debe introducir su número de DNI y la contraseña numérica de home banking de cuatro dígitos para proseguir con el flujo. Una vez capturados esos datos se deberá validar el usuario mediante el **Servicio S1**. De haber algún problema con la validación se debe presentar una pantalla de error al usuario con información relevante.

De ser exitosa la validación, el sistema pasa a la siguiente etapa de presentación de operaciones.

Operaciones
En esta pantalla se le presentarán al usuario las operaciones que podrá acceder el usuario ya validado por la terminal de autoservicio. Las mismas serán los siguientes:

Blanqueo de PIN
El objetivo de la operación es permitir seleccionar y blanquear el PIN de una de las tarjetas pertenecientes al usuario, tanto de las que es titular como de las que son extensiones.
En primera medida el sistema deberá acceder al **Servicio S2** para obtener los productos que el cliente posee en la entidad bancaria. La aplicación deberá presentar como opciones a seleccionar los números de tarjeta y al ser seleccionado uno, el sistema deberá acceder al **Servicio S3**, el cual procederá a blanquear el PIN de la tarjeta elegida.

Saldo de cuenta corriente
El objetivo de este servicio es mostrar al usuario el saldo que posee de la cuenta corriente del cliente en la institución.

El sistema deberá acceder al **Servicio S4** el cual devolverá la información de saldos del cliente. Se deberá presentar dicha información al usuario para su conocimiento.

Últimos movimientos
Este servicio visualizará al usuario los últimos movimientos de su cuenta bancaria. Para ello el sistema deberá interactuar con el **Servicio S5** que devuelve los últimos movimientos de una cuenta de un cliente de la institución.

Flujo general

La aplicación deberá guiar al usuario para realizar las operaciones que requiere, con un formato de pantalla tipo asistente. Ante cualquier error se debe presentar una pantalla describiendo el mismo para su comprensión por el usuario.

Registro de operaciones

El sistema deberá registrar en una Base de Datos todas las acciones que realiza un cliente con la terminal de autoservicio para su posterior análisis. La información que se debe persistir no debe contener datos sensibles, como por ejemplo clave de home banking, saldos, entre otros. Se deberá registrar también el tiempo insumido en la ejecución de las operaciones de la API. La explotación de la información persistida queda fuera del alcance de este requerimiento.

El registro de los datos no debe interferir con la operatoria normal de la aplicación. De mediar algún problema con el acceso al servidor de Base de Datos en el momento del registro, se prefiere obviar dicha persistencia antes que falle el servicio.

Servicios

Para la construcción de la aplicación se utilizarán una API de servicios, que estará expuesta a Internet con el objetivo de lograr la prueba de la mencionada aplicación. Se espera que la implementación de los servicios cambie, no en su comportamiento ni funcionamiento, sino en la tecnología subyacente, **por lo que la aplicación deberá estar preparada ante este cambio.**

Tanto la descripción de los servicios como la forma de interacción se presentará en un anexo.

Requerimientos no funcionales

- La aplicación deberá ser robusta ante cualquier tipo de errores.
- La aplicación deberá ser fácil de usar e intuitiva.
- La interfaz del usuario deberá ser consistente y no tendrá que tener errores de interacción ni de visualización de información.
- La aplicación deberá ser desarrollado sobre la plataforma .NET y en lenguaje C#.
- El programa deberá tener una interfaz gráfica, se sugiere el uso de WinForms integrando los conocimientos y técnicas adquiridos durante la cátedra. La incorporación de conocimientos no adquiridos durante la cátedra serán también bienvenidos.
- El programa deberá persistir las configuraciones y otros datos en una Base de Datos relacional, utilizando un gestor a elección. Se espera que en un futuro puedan configurarse persistencia en distintos gestores de Bases de Datos u otras formas de persistencia (como por ejemplo archivos, Bases de Datos No-SQL, entre otras), por lo que el software debe estar preparado para ello.
- La aplicación deberá contener una bitácora de monitoreo (archivo de log), que permita hacer diagnósticos ante la ocurrencia de errores.
- El código fuente deberá estar correctamente comentado y documentado con los formatos correspondientes.

Entregables

Los entregables del trabajo final son:

- Código ejecutable.
- Código fuente.
- Diagrama de clases y paquetes en formato digital de alguna herramienta UML conocida.
- Guía de instalación y uso del programa.

Importante: Se debe entregar todos los mencionados elementos en un CD cuando se defienda el trabajo.

Evaluación

Se evaluará el uso de buenas técnicas de desarrollo y documentación, legibilidad del código, utilización correcta del lenguaje y finalmente la resolución empleada para satisfacer los requerimientos.

Se deberá realizar una defensa grupal e individual del trabajo en el momento de la entrega,

en donde los alumnos deberán explicar la solución empleada y responder a preguntas de forma individual del equipo docente.

Definición servicios

En esta sección se describen los servicios a emplear por parte de la aplicación. Debido a que no se disponen los servicios reales del core bancario, se plantea el uso de servicios simulados o de tipo Fake. Estos servicios adolecen de seguridad, encriptación, entre otros, y su objetivo es lograr una primera integración operativa con la aplicación.

Los servicios expuestos son de tipo RESTful, que básicamente son request GET HTTP que devuelven datos en formato JSON.

Servicio 1 – Identificación del cliente

El servicio 1 es el encargado de la obtención de información del cliente, para acceder a los datos del cliente el request que se debe formar es el siguiente:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/clients?id=DNI&pass=PASS

Donde:
DNI: es el número de DNI del cliente a identificar.
PASS: es la contraseña del cliente a identificar

Ejemplo:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/clients?id=12345678&pass=1234

Respuesta:

```
[
  {
    "id": 12345678,
    "pass": 1234,
    "response": {
      "client": {
        "name": "Juan Amador",
        "segment": "VIP"
      }
    }
  }
]
```

La respuesta vacía indica que no hay elementos para la consulta, es decir que hubo un fallo en la ejecución del mismo

Tabla de valores de clientes a ser consultados.

DNI	PASS
12345678	1234
12345679	1234
12345680	1234

Un ejemplo de acceso a este servicio a través de un cliente desarrollado en C# se puede acceder en el siguiente repositorio de GitHub

https://github.com/utn-frcu-isi-tdp/tas-db-test-client/blob/master/TAS.Test.Core/Program.cs

Servicio 2 – Obtención de productos del cliente

El servicio 2 es el encargado de la obtención de los productos del cliente, es decir las tarjetas que dispone. Para acceder a los datos del cliente el request que se debe formar es el siguiente:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/products?id=DNI

Donde:
DNI: es el número de DNI del cliente a que se desea obtener los productos.

Ejemplo:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/products?id=12345679

Respuesta:

```
[
  {
    "id": 12345679,
    "response": {
      "product": [
        {
          "number": "1234-0987-4567-9999",
          "name": "Tarjeta Mastercard Banco del Sur",
          "type": "titular"
        },
        {
          "number": "1234-0987-8888-5674",
          "name": "Tarjeta Visa Banco del Sur",
          "type": "extensión"
        }
      ]
    }
  }
]
```

La respuesta vacía indica que no hay elementos para la consulta, es decir que hubo un fallo en la ejecución del mismo

Servicio 3 – Blanqueo de PIN

El servicio 3 es utilizado para blanquear o resetear el PIN de ka tarjeta de débito del cliente.

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/product-reset?number=NUMBER

Donde:
NUMBER: es el número de tarjeta a ser blanqueado el PIN.

Ejemplo:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/product-reset?number=1234-0987-4567-7777

Respuesta:

```
[
  {
    "number": "1234-0987-4567-7777",
    "response": {
      "error": 0,
      "error-description": ""
    }
  }
]
```

Al ejecutar una operación de modificación, la respuesta de la misma tendrá una estructura de código y descripción de error. El código de error 0 (cero) significa que no hubo uno, cualquier otro número significa un error. La respuesta vacía indica que hubo un error irrecuperable.

Servicio 4 – Saldo de cuenta corriente

El servicio 4 informa el estado de la cuenta corriente del cliente. La sintaxis de la consulta es la siguiente:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/account-balance?id=DNI

Donde:
DNI: es el número de DNI del cliente a que se desea obtener el saldo de la cuenta.
Ejemplo:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/account-balance?id=12345680

Respuesta:

```
[
  {
    "id": 12345680,
    "response": {
      "balance": 23900
    }
  }
]
```

La respuesta vacía indica que no hay elementos para la consulta, es decir que hubo un fallo en la ejecución del mismo

Servicio 5 – Últimos movimientos

El servicio 5 devuelve la información de los últimos movimientos de la cuenta corriente del cliente. La sintaxis de la consulta es la siguiente:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/account-movements?id=DNI

Donde:
DNI: es el número de DNI del cliente del que se desean obtener los últimos movimientos de la cuenta.
Ejemplo:

https://my-json-server.typicode.com/utn-frcu-isi-tdp/tas-db/account-movements?id=12345680

Respuesta:

```
[
  {
    "id": 12345680,
    "response": {
      "movements": [
        {
          "date": "2018-12-01",
          "ammount": -500
        },
        {
          "date": "2018-12-01",
          "ammount": -750
        }
      ]
    }
  }
]
```

La respuesta vacía indica que no hay elementos para la consulta, es decir que falló la misma.