

IIC 2113 Diseño Detallado de Software

Actividad 1

Esta actividad debe ser respondida en formato digital.

Guarde en un archivo distinto la solución a cada uno de los enunciados a continuación.

*Incluya un archivo **readme** con los miembros del grupo y los supuestos realizados para el desarrollo de la actividad.*

Comprima todos los archivos en formato zip o rar.

Solo uno de los integrantes del equipo debe subir el trabajo a su buzón.

El incumplimiento de alguna de estas instrucciones será penalizado con 1 punto.

Problema 1: (Diagrama de Estado)

En esta actividad nos enfocaremos en el proceso de realización de transferencias electrónicas bancarias a través de una determinada aplicación móvil para celulares. En primer lugar, la aplicación requiere que los usuarios inicien sesión para acceder al sistema de transferencias. En caso de ingresar erróneamente las credenciales 3 veces seguidas, el acceso se bloquea de forma indefinida (solo se puede restablecer acudiendo a una sucursal para restaurar el acceso). Una vez ingresado al sistema, el usuario debe elegir el destinatario de la transferencia. Si el destinatario no está registrado, el usuario puede agregarlo. Posteriormente, el usuario debe especificar el monto a transferir. El límite estándar para transferir es de 5.000.000 de pesos por día, con la excepción de usuarios registrados hace menos de 24 horas, para ellos el límite es de 200.000 pesos. En caso de exceder el monto, se debe mostrar un mensaje de error y solicitarle al usuario volverlo a intentar. Posteriormente, se debe confirmar la transferencia. Esto puede realizarse ingresando un código Digipass directamente en la aplicación o al recibir confirmación desde otro mecanismo (vía push notification). Si la confirmación es correcta, se confirma la transferencia y se termina el proceso. En caso contrario, se le muestra al usuario un mensaje de error y se le pide reintentar. En caso de 3 errores de confirmación seguidos, se bloquea el acceso de forma indefinida.

Nótese que, si en cualquier momento de este proceso, llegase a transcurrir 1 minuto de inactividad o si el usuario presiona el botón “dormir”, el celular se duerme y el usuario debe desbloquearlo para resumir su actividad desde el mismo punto en que estaba al momento de bloquearse el celular (puede ser a través de código de desbloqueo o huella digital). Si llegasen a transcurrir 5 minutos de inactividad o si el usuario selecciona la opción de “cerrar sesión” dentro de la aplicación, se vuelve a la pantalla de iniciar sesión.

Dibuje un diagrama de estados que permita modelar la situación anterior.

Problema 2: (Diagrama de Clase)

Una empresa de desarrollo de soluciones IoT cuenta con una amplia oferta de productos para satisfacer múltiples necesidades de entusiastas del mundo de Smart Home, incluyendo: luces, televisores, equipos de sonido, cerrojos, refrigeradores, asistentes de voz, cámaras y termostatos. La empresa opera ofreciendo paquetes de productos: dado un cliente interesado, este manifiesta interés en un subconjunto de los dispositivos anteriores, la tienda donde el cliente acude emite un presupuesto (variable según eventuales ofertas) y, si el cliente acepta, un equipo técnico acude al hogar del cliente para instalar todos los dispositivos y configurar una aplicación de control en su Smartphone personal. Desde esta aplicación, el cliente puede controlar remotamente todos los dispositivos inteligentes instalados en su hogar.

La empresa se ve en la necesidad de construir un sistema de gestión de stock y ventas que registre todos los productos vendidos a clientes en sus respectivas sucursales. Toda tienda cuenta con nombre, dirección y un conjunto de empleados. Cada empleado tiene nombre, rut, cargo, dirección, teléfono y email. Toda tienda cuenta con un stock específico de cada producto. Cada vez que se vende un paquete de productos, se le da la oportunidad al cliente de registrarse en la tienda para futuros beneficios. En este caso, se le solicita nombre, rut y email. Si se rehúsa a compartir sus datos, debe figurar en los registros que la venta se realizó a un cliente que prefiere permanecer anónimo.

Toda venta se hace a través de paquetes de productos. Al momento de realizar la venta, debe validarse la disponibilidad del stock. Al concretarse la venta, debe actualizarse el stock de manera acorde. Cada registro de venta debe incluir al vendedor que la cerró, la fecha de la venta, el cliente en cuestión y los productos incluidos en el paquete. Se debe registrar también la dirección de instalación de los productos. La venta debe incluir información de impuestos, flete y costo de servicio de instalación. Cada producto individual debe incluir un código, imagen, descripción y política de garantía. Cada venta debe incluir información de las ofertas vigentes al momento de realizarla. Hay tres tipos de oferta: descuento en un producto particular, N x M (se ofrecen N productos de un tipo por el precio de M) y paquetes de descuento (se ofrece descuento por comprar un conjunto específico de productos juntos). El sistema debe permitir determinar el monto de cada venta mediante el análisis de los productos individuales y las ofertas vigentes al momento de realizar la transacción; y debe poder determinar el historial de ventas de los vendedores para revisar si han cumplido sus cuotas de venta.

Dibuje un diagrama de clases que permita modelar el sistema anterior. Sea exhaustivo al momento de incluir todos los atributos y métodos que estime conveniente. Explícite visibilidad, tipos de los parámetros y tipos de retorno. Incluya información de cardinalidades entre asociaciones, navegabilidad, relaciones de composición y agregación, estructuras jerárquicas e implementación de interfaces.

Problema 3: (Diagrama Global de Interacciones)

Una innovadora aplicación móvil de machine learning permite la ejecución de un pesado modelo de Deep Learning con y sin conexión a Internet. Para ello, se definen 2 modelos con la misma funcionalidad: uno grande, pesado y de gran precisión; y uno pequeño, liviano y de menor precisión. El primer modelo se encuentra hospedado en un servidor. El segundo modelo se encuentra almacenado en el celular. El primer modelo es preferible al segundo dado que es más preciso, más rápido para ser ejecutado (el servidor es mucho más potente que el celular) y no le consume mucha batería al celular. Cuando un usuario requiere ejecutar el modelo, la aplicación revisa si hay o no conexión a Internet. En caso afirmativo, se contacta al servidor, se mandan los parámetros de input del modelo, se ejecuta remotamente el modelo, se guarda el resultado en la BD del servidor, se manda el output de vuelta al cliente y luego se muestra el resultado al usuario. En caso negativo, se ejecuta directamente el modelo liviano en el celular y su resultado se muestra al usuario.

Lamentablemente, debido a la inestabilidad de la red en ambientes móviles, puede ocurrir que luego de haber exitosamente iniciado la ejecución remota del modelo en el servidor, la conexión se pierda y el output nunca se reciba de vuelta. Para paliar esta situación, la aplicación móvil define un timeout T: si después de T segundos de haber iniciado una ejecución remota no se recibe el output de vuelta, se inicia la ejecución local. No obstante, en raras situaciones, puede ocurrir que la respuesta del servidor llegue justo después de T pero antes de que el modelo local termine de ejecutar, en cuyo caso se conserva el resultado del servidor y se aborta la ejecución local.

Dibuje un diagrama global de interacciones que capture toda esta situación.

Problema 4: (Diagrama de Timing)

Se desea construir una aplicación web que genere reportes según solicitudes ingresadas por los usuarios. La generación de reportes es un proceso relativamente intensivo, por lo que resulta muy costoso el implementar una arquitectura con suficientes máquinas virtuales como para construir todos los reportes que se van solicitando en tiempo real. Por lo mismo, se decidió implementar esta aplicación utilizando un sistema de colas y un microservicio aparte para la generación de reportes.

Cuando un usuario ingresa a la aplicación web y genera una solicitud para construir un reporte, la aplicación encola la solicitud en el sistema de colas. El microservicio de generación de reportes solo puede procesar un reporte a la vez. Cuando termina de generar un reporte, le manda un aviso al sistema de colas indicando que se encuentra disponible para recibir un nuevo reporte. En ese momento, el sistema de colas envía la solicitud del nuevo reporte al microservicio. La generación del reporte tarda aproximadamente 10 segundos. Una vez finalizado el reporte, lo guarda en una base de datos (proceso que tarda aproximadamente 2 segundos) y se notifica de vuelta a la aplicación web para que esta a su vez pueda enviar un mensaje al cliente para avisar que el reporte está listo para ser descargado.

Asuma que todo paso de mensajes entre máquinas distintas tarda aproximadamente 1 segundo y que al momento de iniciar el proceso, el microservicio de generación de reportes ya se encontraba trabajando en un reporte previo.

Construya un diagrama de timing que capture el escenario anterior.