



---

**Arquitectura de software - Facultad de Ingeniería**  
**Universidad de Antioquia**

Universidad de Antioquia

Código: 2508585

Versión: 4

Fecha: Agosto 14

---

Tiempo: 4 horas

Examen parcial

Nombre: Johan Daniel Zuluaga Gómez código: C.C. 1001140070

## Caso práctico.

Nubank, el banco digital más grande del mundo con operaciones en Brasil, México y ahora en Colombia con 30 millones de clientes, acaba de comprar un Banco en la ciudad de Sao Paulo denominado **Chubb** para ofrecer servicios de banca completamente digital. Actualmente Chubb cuenta **con 10 millones** de clientes. La meta de Nubank es incorporar los nuevos clientes y **llegar a 50.000.000** de clientes en los próximos 5 años.

### Restricciones

Por políticas de la gerencia, los clientes de NuBank deben quedar registrados en el CRM para dirigir más fácilmente las campañas de mercadeo sobre el CRM Dynamics 365. Uno de los grandes retos de arquitectura consiste en llevar los clientes **de Chubb** a los sistemas de NuBank, garantizando consistencia de la información entre el **CRM Dynamics 365** y el sistema de clientes del banco **Chubb**.

Por restricciones legales se debe garantizar la privacidad en los datos de los clientes, asegurando que la información de los clientes esté protegida mediante SSL.

NuBank está trabajando en un sistema de servicios y bancarización por internet, el cual permitirá a los clientes evaluar entre varios productos de préstamo digital y que puedan realizar el pago en línea de aquel que más se ajuste a sus necesidades.. El sistema debe permitir la venta de productos en todos los países en los cuales tiene presencia NuBank.

Como consideración especial el sistema valida previo a la venta del producto digital de préstamo los siguiente datos:

- Valida que el cliente no se encuentre requerido por INTERPOL mediante microservicio de la agencia de investigación INTERPOLws
- Valida que el cliente no se encuentre reportado a centrales de riesgo mediante un microservicio de DATACREDITO denominado DATACREDITOws
- Con el fin de mitigar el riesgo de tener clientes con dudosa reputación se valida que no se presente lavado de activos ni apoyo al terrorismo mediante un servicio denominado SARLAFTws.

*Usted como arquitecto deberá:*

- Definir un esquema que facilite el gobierno de todas las integraciones que se realizarán
- Soportar el volumen de crecimiento en clientes sin degradar su desempeño
- Especificar como la venta de productos por internet le permitirá de forma flexible cambiar las validaciones y valores sin afectar la disponibilidad de la sucursal virtual.

**Entregables:**

1. Los drivers arquitectónicos (Requisitos funcionales, requisitos no funcionales y restricciones técnicas (20%)
2. Defina cuales son las entidades (clases) arquitectónicamente más significativas (10%)
3. Diagrama de 4+1
  - a. Diagrama de clases (10%)
  - b. Diagrama de componentes (20%)
  - c. Diagrama de paquetes (10%)
  - d. Diagrama de despliegue (10%)



---

**Arquitectura de software - Facultad de Ingeniería**  
**Universidad de Antioquia**

Código: 2508585

Versión: 4

Fecha: Agosto 14

---

**1. Cuáles serían las entradas para la definición de una arquitectura de un proyecto software (10%)**

Lo que necesita un arquitecto para empezar a diseñar la arquitectura de una solución son los requerimientos funcionales y no funcionales, también es muy importante el conocimiento del dominio ya que a veces el cliente no sabe muy bien qué es lo que necesita, y aunque el arquitecto directamente no es analista es el que va a proyectar la especificación en la mejor solución que beneficie a los diferentes stakeholders.

**2. Cuáles son los pasos lógicos para definir una arquitectura de software (5%)**

Luego de tener la especificación y con ello el problema claramente identificado, el arquitecto debe empezar crear la arquitectura, por supuesto, influenciado por previa investigación de casos similares en la industria y su propia experiencia, la arquitectura la puede expresar a través de diagramas donde se aprecie la organización de los diferentes componentes del sistema y su interacción, ya con esto se puede empezar a definir las tecnologías que se usarán, evaluando objetivamente cuál es la más adecuada para el caso, luego de esto se puede ahondar en los detalles de la implementación.

**3. Defina la diferencia entre un patrón de diseño y un patrón arquitectónico (5%)**

Un patrón de diseño como abstract factory es más cercano a la implementación aunque también establece un orden e interacción componentes pero en un nivel más bajo, en cambio un patrón arquitectónico como el de microservicios es más alto nivel que también permite cierta organización del software, es decir si el de diseño se trabaja dentro de un paquete en el arquitectónico se trabaja en la organización de los paquetes o módulos en sí.

## 1. Drivers arquitectónicos:

### Requisitos funcionales:

- CRUD de **clientes**.
- Los clientes tienen asociado un producto de **crédito** digital.
- Los clientes pueden **pagar** en línea el **crédito** elegido.
- Se debe validar que el cliente:
  - No se requiera por INTERPOL.
  - No se encuentra reportado en centrales de riesgo.
  - No realiza lavado de activos ni apoyo al terrorismo.

### Requisitos no funcionales:

- Escalable.
- Flexible.
- Siempre disponible.
- Segura.

### Restricciones técnicas:

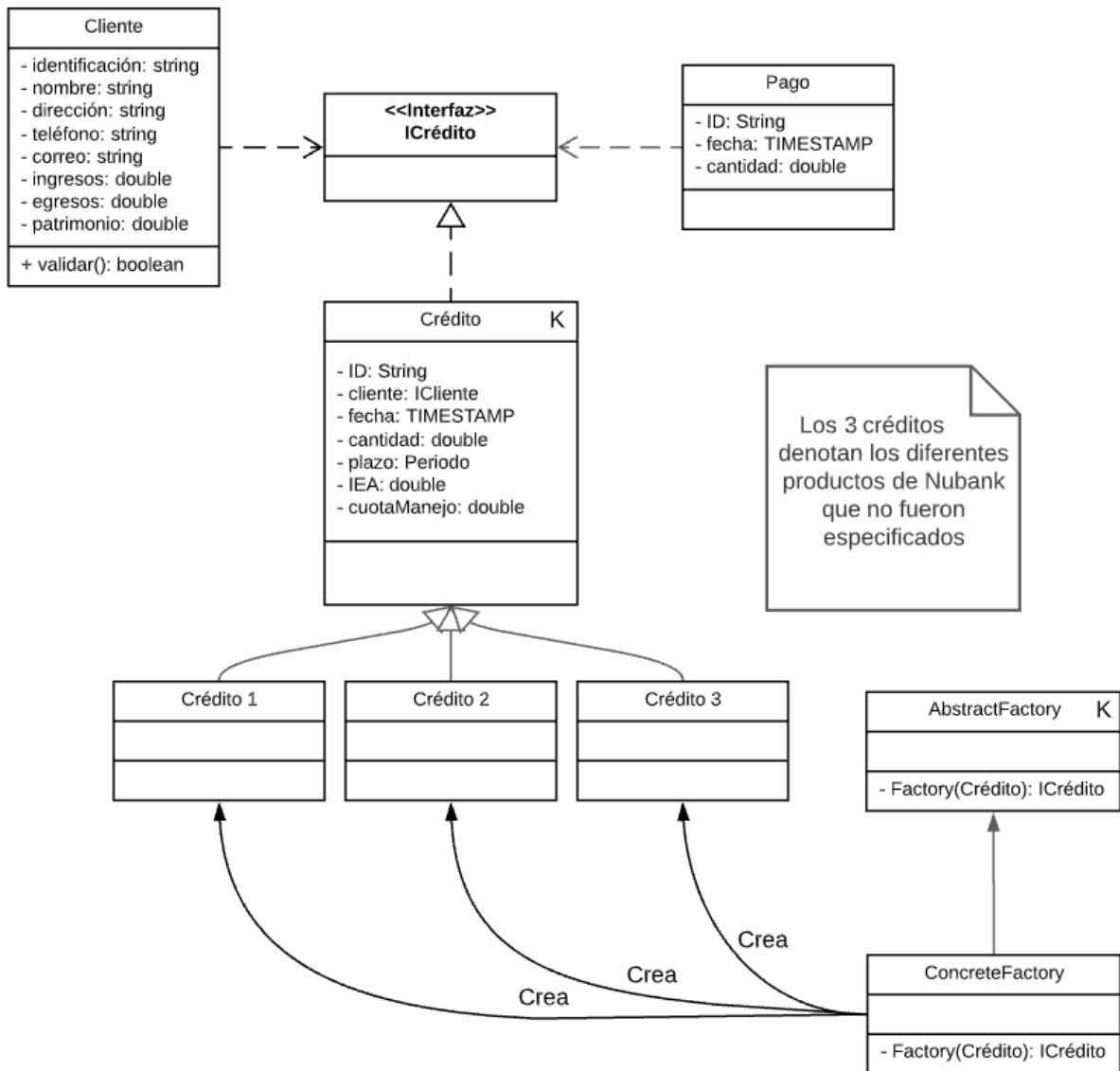
- Integrar la base de datos de NuBank con la de Chubb
- Para la seguridad se debe usar el protocolo SSL.
- Establecer conexión con las APIs de consulta para la validación de los clientes: INTERPOLws, DATACRÉDITOWs, SARLAFTws.
- 

## 2. Clases arquitectónicamente más significativas:

Cliente, Préstamo, Pago.

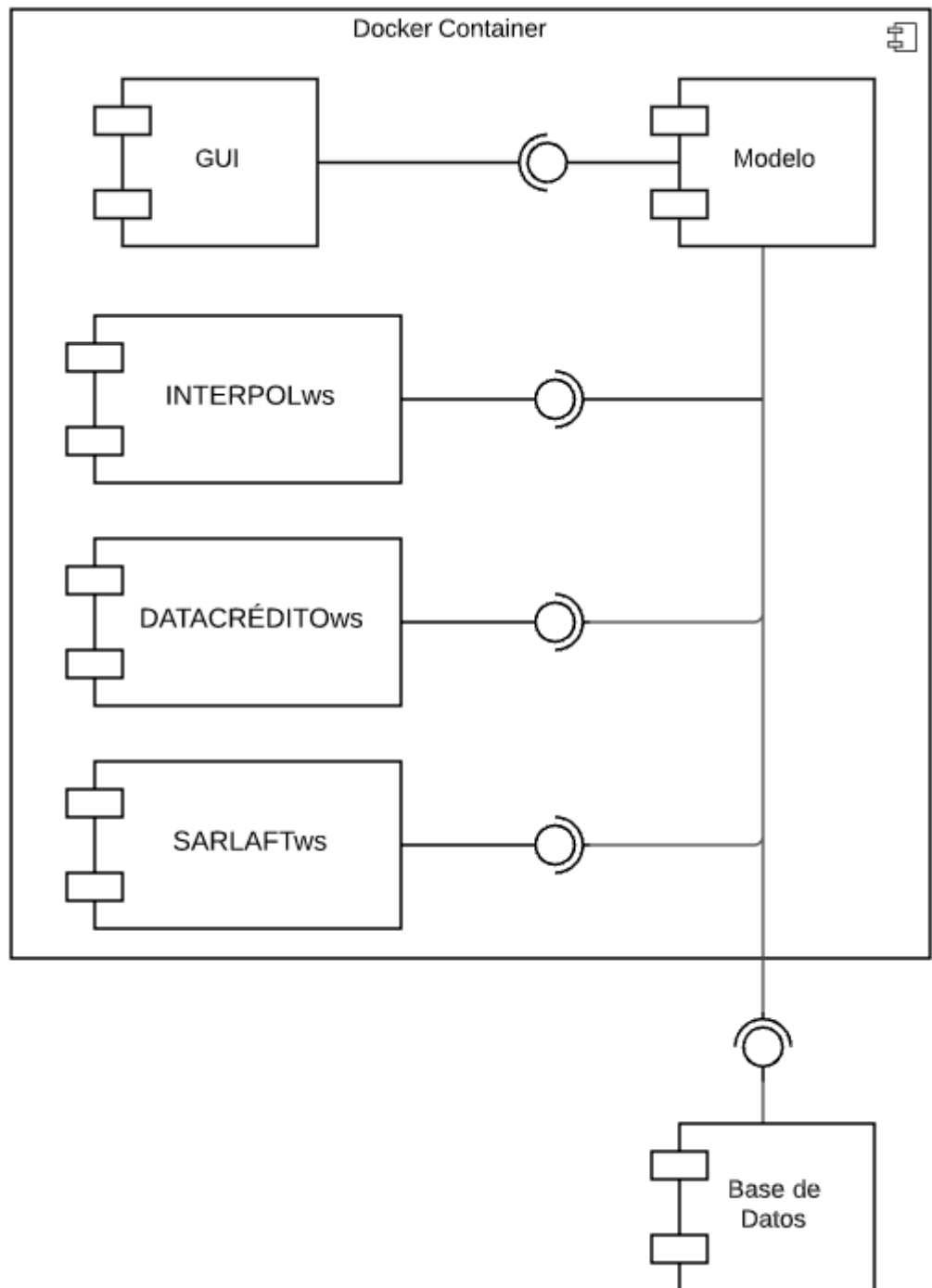
## 3. Diagramas 4 vistas:

a. Diagrama de clases del modelo:

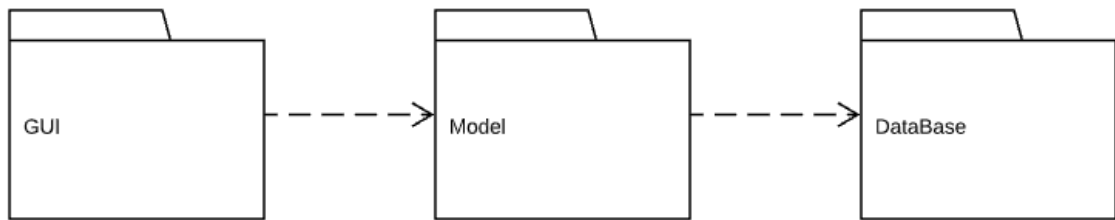


b.

b. Diagrama de componentes:



c. Diagrama de paquetes:



d. Diagrama de despliegue:

