



JULIO ANTHONY ENGELS RUIZ COTO – 1284719

# Análisis y diseño II

## Parcial II

### Reto I

#### Propuesta de estilo arquitectónico

Dado el siguiente problema, proponga la arquitectura de la solución de software idónea y justifique su respuesta. Utilice la hoja adjunta como referencia.

1. Identifique las características críticas.

**Seguridad:** ya que el sistema se tiene que saber proteger ante cualquier ataque inminente y no poner en riesgo los datos de los estudiantes.

**Escalabilidad:** debe ser capaz el sistema para manejar la carga de cientos de estudiantes.

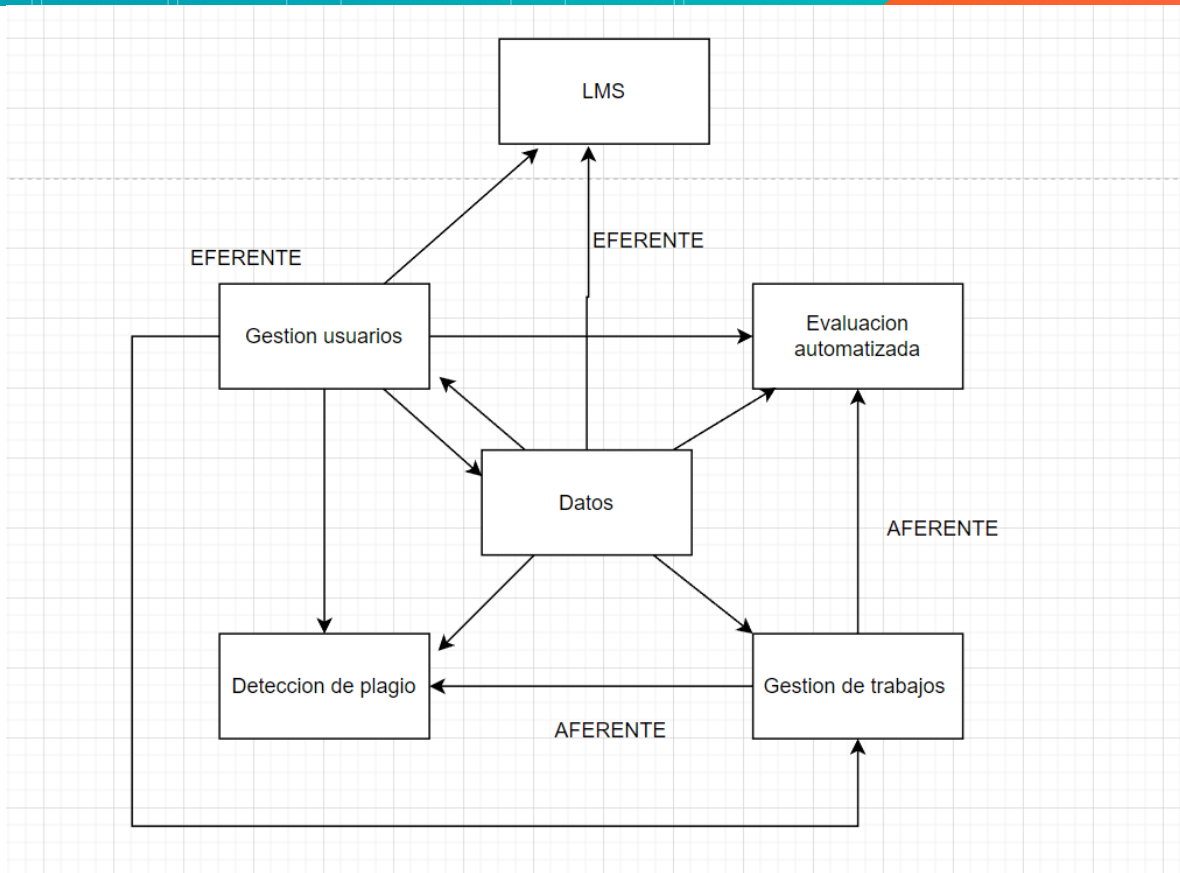
**Usabilidad:** el sistema tiene que ser de un uso fácil para el personal administrativos como para los estudiantes.

**Auditoria:** el sistema debe tener registros auditables de las notas.

**Fiabilidad:** ya que como menciona la calificación automática y también la detección de plagio estas a su vez deben ser precisas y confiables.

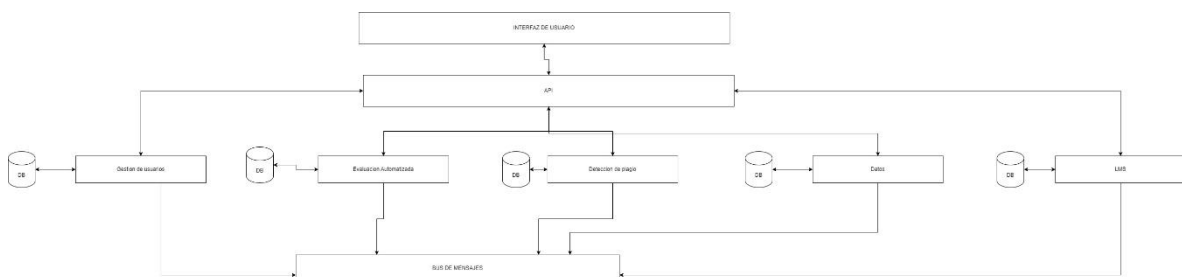
**Resiliencia:** ya que en cualquier circunstancia de algún fallo este debe de poner recuperarse para minimizar el tiempo de interrupción o de caída.

2. Diseñe un bosquejo de los componentes de software, sus dependencias y tipos de acoplamiento.



3. Diseñe la arquitectura con base a los estilos que considere conveniente, justifique por qué eligió el(los) estilo(s). Utilice la plantilla para justificarlo (architecture-styles-worksheet).

En el presente diagrama hago un pequeño bosquejo de la arquitectura de microservicios ya que considero que la escalabilidad es ideal para sistemas con distintos componentes ya que estos necesitan ser escalados de una manera independiente.





### Check Your Work

A university has greatly expanded its CS course and wants to be able to automate the grading of simple programming assignments.

- Users: 300+ students per year, plus staff and admin.
- Requirements:
  - students must be able to upload their source code, which will be run and graded
  - grades and runs must be persistent and audit-able
  - required plagiarism detection system involving comparing with other submissions and also submitting to a web-based service (TurnItIn).
  - integration required with the University's learning management system (LMS)
  - professor sets due date and time, after which submissions are rejected
  - students can submit as many attempts as they want to improve their grade
  - professors determine grading criteria, which may include metrics and/or tests
- Additional Context:
  - University's LMS is mainframe based and quite difficult to make changes to
  - grades are audited each year by state-based regulatory body
  - University has very little budget for IT as it is building a spare stadium for SportsBall
  - University has a record for highest-performing CS graduates in the country

## Reto II

### Auditoría y asesoría en arquitectura de software

Le han pedido su apoyo para identificar y analizar los siguientes elementos de la arquitectura de varios sistemas de información. El departamento de TI fue desintegrado y no se encontró documentación sobre la arquitectura, bajo la premisa que su estilo de trabajo era "ágil".

1. ¿Es válido que al trabajar con modelos de software ágiles no exista la documentación de la arquitectura del sistema?

*Cuando se tiene un enfoque agile sabemos que la entrega va a ser incremental con una colaboración estrecha entre los miembros del equipo más sin embargo esto no quiere decir o hace mención que no se deba tener documentación en absoluto quiere decir que la documentación no debe ser un impedimento en las entregas de software. Se sabe que dicha documentación es de suma importancia para la toma de decisiones y en mantener la coherencia a medida que dicho sistema evolucione. Por lo tanto, el enfoque ágil favorece en el software funcional sobre la documentación, es de mucha ayuda tener suficiente documentación ya que con ello se logra entender la arquitectura y el diseño del sistema que se realizara.*

2. Se encontró este fragmento de documento. ¿A qué corresponde?

*El fragmento de documento corresponde a una nueva especificación que se tendrá para el método del servicio web WS003, este método se llamara activartarjetacredito , se observa también que este método tendrá los datos de entrada y de salida que se aceptaran y los que devolverán, todo esto esta descrito un formato de XML. Se observa que el output de dicho método tiene un HRESULT lo que indica el resultado de la operación es exitosa o no, esta da un mensaje que describe el resultado y un numero de autorización, y por ultimo las premisas nos brinda cierta información sobre el contexto que debe usarse este servicio. Este*

documento también se puede interpretar ya sea para un caso de uso o para un caso de prueba dependiendo del contexto en el que se encuentre.

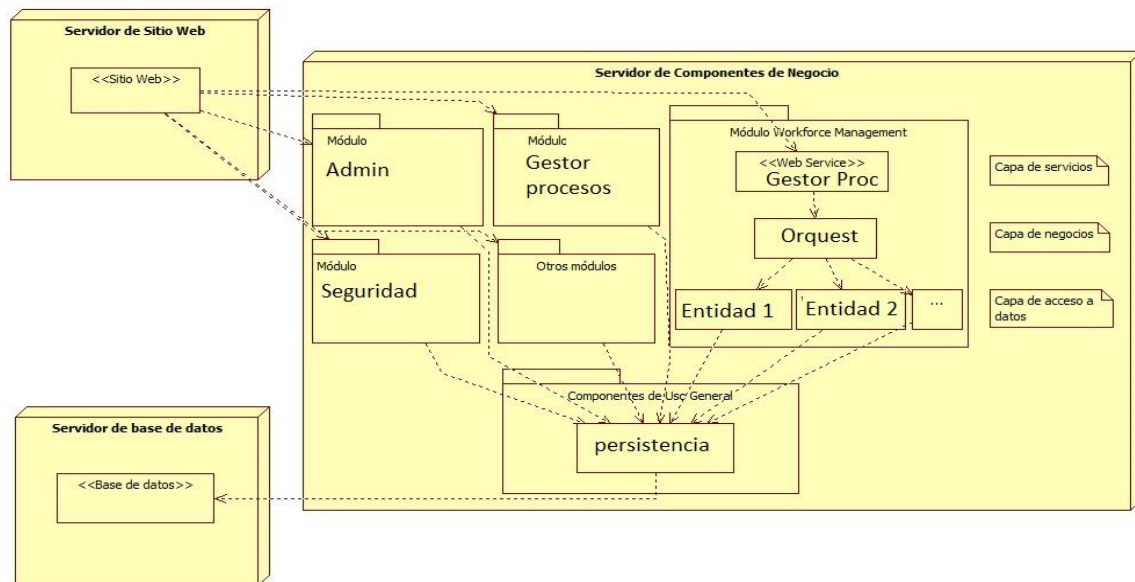
Requerimiento: se agregarán al servicio web WS003 los siguientes métodos.

Proceso / método	INPUT	OUTPUT
ActivarTarjetaCredito	Código Número de tarjeta DPI Fecha de nacimiento Nit	Hresult Número de autorización
<i>Descripción: activa la tarjeta de crédito especificada.</i>		
<pre>&lt;?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?&gt; &lt;WS003.INPUT&gt;   &lt;codigo&gt;1010&lt;/codigo&gt;   &lt;numero-tarjeta&gt;4290443321100101&lt;/numero-tarjeta&gt;   &lt;DPI&gt;2485055520203&lt;/DPI&gt;   &lt;fecha-nacimiento&gt;     &lt;anio&gt;1995&lt;/anio&gt;     &lt;mes&gt;4&lt;/mes&gt;     &lt;dia&gt;11&lt;/dia&gt;   &lt;/fecha-nacimiento&gt;   &lt;nit&gt;3336988&lt;/nit&gt; &lt;/WS003.INPUT&gt;</pre>		
<pre>&lt;?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?&gt; &lt;WS003.OUTPUT&gt;   &lt;Resultado&gt;     &lt;HRESULT&gt;1&lt;/HRESULT&gt;     &lt;mensaje&gt;Activación realizada con éxito.&lt;/mensaje&gt;   &lt;/Resultado&gt;   &lt;area-datos&gt;     &lt;numero-autorizacion&gt;456789206&lt;/numero-autorizacion&gt;   &lt;/area-datos&gt; &lt;/WS003.OUTPUT&gt;</pre>		

#### Premisas

- Para activar tarjetas de crédito y débito el cliente no necesariamente debe tener un usuario virtual.
- Para activar tarjetas de crédito y débito el cliente no necesariamente ha iniciado sesión.
- Todas las operaciones de activación deben dejar un registro en la bitácora del orquestador.

3. Este diagrama. ¿Qué significa? ¿Qué tipo de diagrama es? ¿Cómo se lee e interpreta?



Por lo que se observa, es un diagrama de arquitectura de software básicamente nos brinda una descripción de la estructura general del sistema y como los distintos componentes que se observan del sistema están interactuando. Yo considero que la arquitectura que se está viendo en esta imagen es de la arquitectura de N-capas ya que puedo relacionar el servidor de sitio web con la capa de presentación, el servidor de componentes de negocio con la capa de negocio y el servidor de base de datos con la capa de datos y el componente de persistencia. Si realizamos un poco más de inspección o de zoom en el servidor de componente de negocio se puede observar que existe una arquitectura basada en componentes así también la presencia de servicio web y de orquestación lo que conlleva a una arquitectura orientada a servicios como se conoce SOA para este componente del sistema.

El diagrama se lee , existe un front-end que los representa el servidor de sitio web, desde este servidor se siguen las flechas hasta el servidor de componentes de negocio lo que conlleva a que hace las solicitudes del usuario se pasan de la interfaz de usuario a la lógica de negocio del sistema cada módulo proporciona diferentes funcionalidades, así también se observa una subdivisión en diferentes capas lo que conlleva a una separación de responsabilidades dentro de dicho servidor. En cuanto al web service y el Orquest como ya se mencionaba hace referencia a la arquitectura SOA y sus respectivos tipos de datos o entidades con la que el sistema hace su labor, todo esto conecta con unas flechas al componente



*de uso general llamado persistencia eso se entiende que los datos se almacenan y recuperan de la base de datos por medio del componente de persistencia.*