Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

EVALUACIÓN	Obligatorio	**GRUPO**	N7A	**FECHA**	26-set-2018
MATERIA	Arquitectura de Software				
CARRERA	ID-AN				
CONDICIONES	- Puntos: Máximo: 35 Mínimo: 15 - Fecha máxima de entrega: 29-nov-2018 LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP, RAR O PDF. **IMPORTANTE:** - Inscribirse - Formar grupos de hasta dos personas. - Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: “RECORDATORIO”				

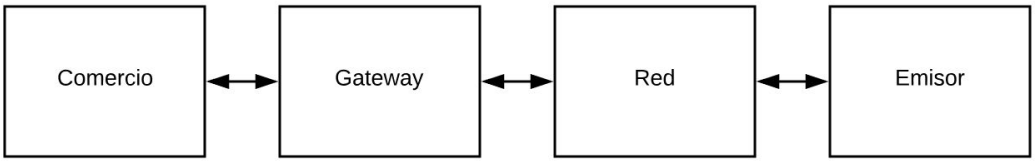
Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Índice

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Contexto del problema

El mercado de medios de pago es altamente competitivo y presenta grandes desafíos para la implementación de soluciones seguras, robustas y flexibles.



Cuando una persona quiere obtener una tarjeta de crédito, se dirige a una entidad financiera habilitada para ser **emisor** (usualmente bancos). Cada una de estas entidades ofrece una cartera de productos diferente que se adapta a las necesidades del cliente y su poder adquisitivo. Por ejemplo, el Banco Santander Uruguay emite tarjetas Visa Standard y Gold, Mastercard Platinum y Titanium, etc.

Cuando un **comercio** quiere realizar ventas con tarjetas de crédito, debe solicitar los datos de la tarjeta de crédito al cliente y de alguna manera comunicarse con el emisor para verificar si la tarjeta es válida, vigente, tiene saldo, etc. Para esto, debería realizar distintas integraciones directas con cada entidad emisora del país o región.

Dada la complejidad y el gran número de integraciones que un comercio debería realizar, se creó un nuevo nicho de mercado que algunas empresas llamadas **gateways** comenzaron a explotar. Su trabajo es unificar el servicio de pagos hacia un comercio, resolviendo en su nombre las integraciones para aceptar los distintos medios de pago. Es decir, como comercio me integro una sola vez con MercadoPago, pero a través de ellos puedo aceptar tarjetas Visa Santander, Mastercard Itaú, etc.

A su vez, existen las **redes** (Visa, Mastercard, American Express), cuyo rol es actuar de intermediarios entre quienes quieren realizar transacciones de pago (comercios y gateways) y quienes los aprueban (emisores). Entre otras cosas, las redes controlan la veracidad de cada transacción, comparando el historial de actividad de cada persona, el país de emisión de la tarjeta, etc. Por más que existe un marco global, cada red provee servicios distintos dependiendo de la región y a veces del país, lo que provoca que un gateway deba hacer una integración distinta para cada representante de Visa por ejemplo en las regiones donde quiere operar.

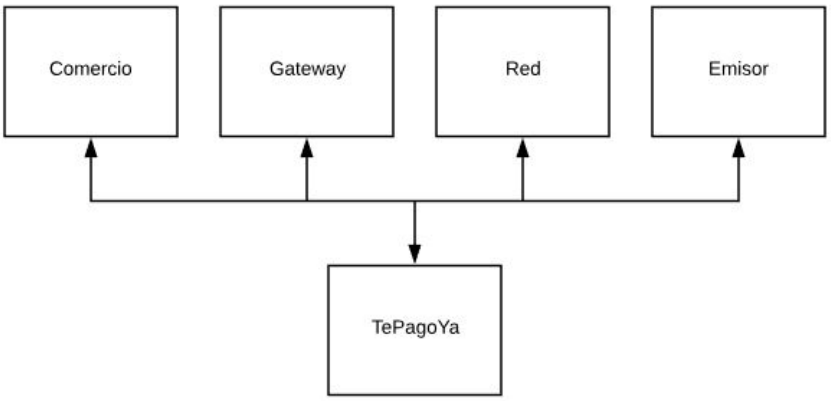
En la realidad global de hoy en día, especialmente en América Latina, cuando un comercio quiere aceptar medios de pago online, es muy difícil encontrar un solo gateway que le provea una integración con todas las redes y todos los emisores por un costo y condiciones adecuadas. Por lo tanto, se termina creando un complicado árbol de comunicaciones desde el comercio a varios gateways, cada gateway a varias redes y cada red a varios emisores.

Para complicar aún más la situación, existen muy pocos estándares y reglas (PCI, ISO, etc) sobre cómo deben comunicarse estos actores. Cada uno desarrolla sus servicios de la manera que decide y quien los consume debe adaptarse al mismo. Para citar un simple ejemplo, las siguientes son distintas maneras de manejar un dato de “fecha de transacción”: “04/10/2018”, “10/04/2018”, “2018/10/04 1:15pm”, “2018-10-04 13:15:56.555”, “2018-10-04T13:15-0300” y “1538658959”

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Propuesta de desarrollo

Teniendo en cuenta el complicado universo de los pagos online, surge la urgente necesidad de simplificar lo más posible la integración de los distintos sistemas. La propuesta es crear una herramienta que provea la funcionalidad de actuar de intermediario entre los cuatro diferentes actores (comercio, gateway, red y emisor), resolviendo la mayor parte posible de los desafíos de integración como pueden ser los diferentes formatos de fechas y números, campos obligatorios y opcionales, formatos de representación de datos, etc.



Este sistema deberá poder integrarse con cada uno de los actores, permitiendo que la comunicación sea siempre hacia y desde el nuevo sistema, indicando de alguna manera cuál es el actor de destino de cada comunicación. Imaginemos el siguiente escenario para ejemplificar:

- Gateway A quiere comunicarse con la Red B y la Red C
- La Red B quiere recibir las fechas en formato “yyyy-MM-dd” y la Red C quiere recibirlas como “dd/MM/yy”
- Para poder comunicarse con cada uno, A deberá implementar lógica para convertir sus fechas (originalmente en formato “MM_dd_yyyy”) al formato esperado por B y C en cada caso.

Con el nuevo sistema (S) a desarrollar, A puede enviar la fecha en un solo formato (definido por S) y será S quien se encargue de comunicarse con B y C convirtiendo la fecha al formato esperado por cada uno. De esta manera, A puede despreocuparse y pasar la responsabilidad de adaptación hacia S.

Es importante tener en cuenta que:

- Cada aplicación se desarrolla, distribuye y ejecuta de forma independiente
- Cada aplicación gestiona su propio repositorio de datos (incluso pueden ser de diferente tipo: RDBMS, NoSQL, file system, etc.) y por lo tanto no hay “esquemas de datos” compartidos entre aplicaciones a nivel de persistencia

En su rol de arquitecto de software, se le encomienda diseñar, construir y demostrar prototipos funcionales del backend de las aplicaciones comercio, gateway, red y emisor interactuando a través del mecanismo de integración TePagoYa.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Requerimientos

REQ-1: Compra en Comercio

Un consumidor puede realizar una transacción de compra en un **comercio**, informando los siguientes datos:

- Tarjeta
  - Número
  - Vencimiento
  - Nombre de titular
  - Código de seguridad
- Dirección de Envío
  - Calle
  - Número de puerta
  - Ciudad
  - País
  - Código postal
- Dirección de Facturación (mismos datos que la de Envío)
- Monto de transacción
- Fecha de transacción
- Producto a comprar
  - Cantidad
  - Nombre
  - Categoría

En función de la categoría del producto a comprar, el comercio deberá seleccionar el **gateway** que procesará la compra y enviarle la solicitud. Por ejemplo, los electrodomésticos pueden ser procesados por el Gateway A y los alimentos por el Gateway B. Esto debe ser configurable sin necesidad de modificar el código de los comercios.

La respuesta hacia el consumidor debe informar si la transacción fue exitosa o no y un código de transacción en caso de necesitar devolverla eventualmente.

REQ-2: Compra en Gateway

Cuando un **comercio** envía la información pertinente para que el **gateway** procese un pago, este debe decidir a qué **red** enviar el pago según a cuál de ellas pertenece la tarjeta. La identificación de la red debe realizarse en tiempos menores a 10 ms en promedio bajo cargas de 1000 solicitudes por minuto.

REQ-3: Compra en Red

Cuando un **gateway** envía una transacción a la **red**, esta realiza los controles de prevención de fraude y en caso de que pase, la envía al **emisor** para el paso final de aprobación. Para el análisis de fraude, se revisa la cantidad de transacciones que hizo esa tarjeta en las últimas 72 horas y si supera un límite determinado,

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

la transacción se rechaza. Este límite debe poder ser alterado en tiempo de ejecución sin necesidad de interrumpir la operación de la red.

La identificación del emisor de la tarjeta debe cumplir con las mismas condiciones que para la identificación de la red.

REQ-4: Compra en Emisor

Cuando la **red** envía una transacción a ser aprobada en el **emisor**, este debe confirmar que la tarjeta sea válida y el saldo de la cuenta del titular sea suficiente para el monto a aprobar. Para que una tarjeta sea válida, debe haber sido emitida por el emisor, pasar el algoritmo de Luhn, no estar vencida, ni bloqueada (por falta de pago), ni denunciada (por robo o pérdida).

REQ-5: Devolución

Deberá ser posible realizar la devolución de cualquier compra realizada previamente que no haya superado el límite de días definido en el **emisor**. Realizando todos los controles pertinentes, notificando y asegurando la consistencia de todas las aplicaciones involucradas.

REQ-6: Chargeback

El usuario puede realizar un desconocimiento de compra en su **emisor**. El emisor deberá informar al **comercio** de este evento para que ejecute las acciones administrativas pertinentes. Se debe tener en cuenta que el comercio puede llegar a demorar un tiempo significativo en encontrar la transacción original ya que se puede hacer un chargeback de hasta 6 meses atrás.

REQ-7: Cierre de Lotes

El **comercio** le solicita diariamente al **gateway** la transacción de cierre de lotes, en donde se informa al máximo detalle los movimientos de dinero que hubo en su cuenta en ese día (cerrando a determinada hora acordada). Esta transacción debe responder en un tiempo promedio menor a 50 ms bajo cargas de 5000 rpm.

REQ-8: Cumplimiento de PCI

Toda aplicación debe cumplir con los estándares de seguridad de PCI al máximo nivel posible, pudiendo demostrar y justificar a cada momento el nivel de cumplimiento.

REQ-9: Mecanismo de integración de aplicaciones (TePagoYa)

Desarrollar la aplicación **TePagoYa** para resolver la integración e interoperabilidad de las aplicaciones actuales y futuras. Toda aplicación que se registra en TePagoYa puede actuar como prestador de servicios y/o consumidor de servicios.

Para cada prestador de servicios se debe poder:

- Registrar una “interfaz de servicio” que permita acceder a su funcionalidad.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

- Especificar propiedades sobre la forma de usar cada interfaz (por ejemplo, el formato de las estructuras de datos que se intercambian por la interfaz, o el tipo de comunicación).

Para cualquier consumidor de servicios de debe poder:

- Localizar una interfaz de un servicio registrado
- Facilitar la invocación de las funcionalidades del servicio a consumir

REQ-10: Gestión de errores y fallas

El sistema debe proveer suficiente información, de alguna forma, que permita conocer el detalle de las tareas que se realizan. En particular, en el caso de ocurrir una falla o cualquier tipo de error, es imprescindible que el sistema provea toda la información necesaria que **permita a los administradores hacer un diagnóstico rápido y preciso sobre las causas**. Se espera que la solución contemple la posibilidad de **poder cambiar las herramientas o paquetes concretos** que se utilicen para producir esta información, así como **reutilizar esta solución en otras aplicaciones**, con el menor impacto posible en el código.

REQ-11: Autenticación de operaciones

Actualmente, cualquier usuario con acceso a la red informática donde están conectados los servidores de aplicaciones puede invocar libremente cualquier operación expuesta por las API de backend de cualquier aplicación (imagine un usuario ejecutando llamadas http usando Postman o cualquier otra herramienta).

Para toda aplicación “prestadora de servicios” que se registre en **TePagoYa**, se requiere que todas sus operaciones publicadas deben ser invocadas únicamente a través de **TePagoYa**. Es decir que toda invocación a una operación de un servicio registrado que se realice por fuera de **TePagoYa** se debe considerar como no legítima.

De igual forma, toda invocación a **TePagoYa** debe provenir de una aplicación “consumidora de servicios” legítima. Por ejemplo, realizar una invocación http por Postman a TePagoYa para consumir un servicio debe considerarse como un acceso no legítimo.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Condiciones generales del trabajo obligatorio

1. Restricciones

- La implementación del backend debe desarrollarse en NodeJS utilizando las tecnologías vistas en el curso. Es opcional y abierta la elección de packages que puedan ayudar al desarrollo, teniendo que justificar la elección en la documentación.
- Las APIs expuestas por los servicios de comercios deben ser REST. Para las comunicaciones entre otros servicios, se tiene libertad de seleccionar el mecanismo.
- Todo el código fuente, documentación, archivos de configuración o cualquier otro artefacto referido al desarrollo de los prototipos debe gestionarse en el repositorio Git asignado.

2. Objetivo del trabajo

El objetivo de este obligatorio es:

- Diseñar y construir una arquitectura de software que pueda ser demostrada en la defensa.
- Producir un documento de arquitectura que sirva para explicar las estructuras diseñadas en función de los atributos de calidad relevantes que se hayan identificado.

Se pueden proveer datos de prueba precargados en la base de datos con el fin de facilitar la demostración.

3. Entregables

- Documento con la descripción de la arquitectura diseñada siguiendo el modelo **Views&Beyond** tratado en el curso. **Prestar especial atención** a los criterios de corrección detallados más adelante.
- Código fuente en repositorio Git con la implementación de los requerimientos solicitados.
- **[opcional]** Guías de instalación, configuración o uso, archivos de configuración o datos, para poder ejecutar la aplicación (pueden facilitarse también en el mismo repositorio Git del código).

4. Criterio de corrección

La evaluación del obligatorio se divide en aspectos teóricos y prácticos. Para cada uno de ellos se aplican determinados criterios de corrección. A continuación se muestra la distribución de puntajes para cada aspecto, seguida del detallado de cada aspecto.

Funcionalidad	30%
Requerimientos no funcionales	25%
Documentación	20%
Calidad del diseño	10%
Calidad del código y control de versiones	10%
Demostración al cliente	5%
TOTAL	**100%**

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

4.1. Funcionalidad

En este apartado, se identifican los siguientes criterios de corrección:

- Se implementaron sin errores todos los requerimientos funcionales propuestos.
- Se implementaron todos los requerimientos funcionales propuestos pero se detectan errores menores que no afectan el uso normal del sistema.
- Se implementaron los principales requerimientos funcionales, aunque no todos, pero se detectan errores menores que no afectan el uso normal del sistema.
- Se implementaron los principales requerimientos funcionales, aunque no todos, pero se detectan errores que afectan el uso normal del sistema.
- Los requerimientos funcionales implementados son básicos y/o se detectan errores que afectan el uso normal del sistema.

4.2. Requerimientos no funcionales

Para cada RNF propuesto o identificado se aplican las siguientes prioridades en la corrección:

- Se implementaron satisfactoriamente tácticas adecuadas para cada RNF y se verifica su funcionamiento.
- Se implementaron las tácticas adecuadas al RNF pero no se verifica su funcionamiento.
- Se implementaron parcialmente las tácticas adecuadas al RNF.
- Las tácticas implementadas no son las adecuadas para el RNF.
- No se implementó ninguna táctica para satisfacer el RNF.

4.3. Documentación

Una documentación aceptable cumple con lo siguiente:

- Incluye al menos una vista de tipo Módulos, una de tipo Componentes y Conectores y una de tipo Asignación.
- Cada vista está documentada en base a la guía vista en el curso (representación primaria, catálogo de elementos, justificaciones de diseño, guías de variabilidad)
- Los diagramas se realizan en UML 2 y su uso es correcto.
- Las justificaciones de las decisiones de diseño tomadas son relevantes desde el punto de vista arquitectónico y permiten entender el “por qué” del diseño (y no el “qué hace” o el “cómo está hecho”).

4.4. Calidad de diseño

Un diseño aceptable cumple lo siguiente:

- La paquetización del código representa la descomposición lógica (módulos) de la aplicación.
- Las estructuras y comportamientos documentados sirven como guía para la comprensión del código implementado.
- Las tácticas y patrones arquitectónicos se implementan correctamente a partir de su objetivo y teniendo en cuenta sus ventajas y desventajas para favorecer o inhibir atributos de calidad.
- Se cumple con el Principio de Responsabilidad Única (SRP).
- Se cumple con el Principio de Dependencias Acíclicas (ADP).
- Se gestionan los posibles errores en todos los casos.

4.5. Calidad del código y control de versiones

Un código de calidad aceptable cumple lo siguiente:

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

- Cumple las convenciones de codificación de NodeJS (https://docs.npmjs.com/misc/coding-style)
- La gestión del código del obligatorio debe realizarse utilizando el repositorio Git de Github, apoyándose en el flujo de trabajo recomendado GitFlow (nvie.com/posts/a-successful-git-branching-model).
- Luego de la entrega se le debe dar acceso a los docentes al repositorio. (solo es necesario darle acceso a los dos docentes de su dictado)

4.6. Demostración al cliente

Cada equipo deberá realizar una demostración al cliente de su trabajo. Dentro de los aspectos que un cliente espera se encuentran:

- Ver la demo cuando él esté listo y no tener que esperar a que el equipo se apronte.
- Probar la solución y que la misma funcione sin problemas o con problemas mínimos, y de buena calidad de interfaz de usuario.
- Conocer las capacidades de cada uno de los integrantes del equipo, pudiendo preguntar a cualquier integrantes sobre la solución, su diseño, el código y sobre cómo fue construida, y así apreciar que fue un trabajo en equipo. Todos los integrantes deben conocer toda la solución.
- Verificar el aporte individual al trabajo por parte de cada uno de los integrantes del equipo y en función de los resultados, se podrán otorgar distintas notas a los integrantes del grupo. Se espera que cada uno de los integrantes haya participado en la codificación de parte significativa del obligatorio.

Esta demostración al cliente hará las veces de defensa del trabajo.

NOTA: El incorrecto funcionamiento de la instalación puede significar la no corrección de la funcionalidad. En el caso de la demostración al cliente en el laboratorio, cada grupo contará con 15 minutos para la instalación de la aplicación. Luego de transcurridos los mismos el cliente comenzará a molestarse impactando en su percepción sobre el trabajo. A su vez el cliente podrá realizar preguntas a cualquiera de los integrantes las cuales podrán afectar la valoración que hace de cada integrante del equipo.

UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

➢ **Obligatorios** (Cap.IV.1, Doc. 220)

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. La entrega se realizará desde **gestion.ort.edu.uy**
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación. S**ugerimos realizarlo con anticipación.**
3. **Uno de los integrantes del grupo de obligatorio será el administrador del mismo** y es quien formará el equipo y subirá la entrega
4. Cada equipo debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar)
5. El archivo a subir debe tener **un tamaño máximo de 40mb**
6. Les sugerimos **realicen una 'prueba de subida' al menos un día antes**, donde conformarán el **'grupo de obligatorio'.**
7. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
8. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc)
9. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor pasar por la oficina del Coordinador o por Coordinación adjunta **antes de las 20:00hs.** del día de la entrega

Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o Secretario Docente.