

PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA  
FACULTAD DE CIENCIAS E INGENIERÍA  
ESCUELA DE INGENIERÍA EN COMPUTACIÓN Y TELECOMUNICACIONES

Programación Web Avanzada  
**Proyecto Final**

**Objetivos:**

- Utilizar las herramientas aprendidas durante el semestre.
- Aplicar una arquitectura de microservicios.
- Aplicar un esquema de alta disponibilidad con balanceadores.
- Uso de Correo Transaccionales.

**Desarrollo del proyecto.**

El estudiante realizará cada uno de los ejercicios indicando en dicho documento interactuando con el servidor de versiones. Las practicas serán utilizando el IDE de su preferencia; los fuentes serán gestionados en el control de versiones Github y la práctica puede ser realizada en grupo de dos.

**Forma de Entrega:**

La entrega de la práctica se debe realizar un reporte con lo trabajado, **conjuntamente con una presentación del proyecto en (10 minutos) con un vídeo bajo la plataforma youtube con su respectivo reporte**; la presentación debe cubrir cada unos de los requerimientos solicitado en el proyecto trabajado. La práctica será entregada según la fecha indicada en la PVA.

## Desarrollo del proyecto

Una empresa reconocida de Santiago se entera que tenemos mucha experiencia en el desarrollo de sistemas web; dicha empresa se comunica con nosotros y nos plantea la necesidad de centralizar todas sus operaciones en un sistema informático que sea de rápido acceso para sus clientes y diseñado para garantizar una alta disponibilidad.

La empresa se dedica a realizar servicio de multimedia de eventos (fotografías y vídeos), los cuales tiene unos planes específicos para que los clientes lo seleccionen, dichos planes son:

- Pre-Boda, monto de RD\$1,000.00
- Boda, monto de RD\$5,000.00
- Cumpleaños, monto de RD\$3,000.00
- Vídeo de evento, monto RD\$4,000.00

La Empresa requiere una mejora de sus procesos para automatizar los pedidos, el pago y la asignación de los trabajos a los fotógrafos, camarógrafos y personal de apoyo de los trabajos contratados. Una vez el cliente selecciona el paquete de servicio, se requiere que pueda realizar el pago electrónico y genere una solicitud para que los empleados puedan asignarse el trabajo y posteriormente completarlo, manteniendo en todo momento al cliente notificado.

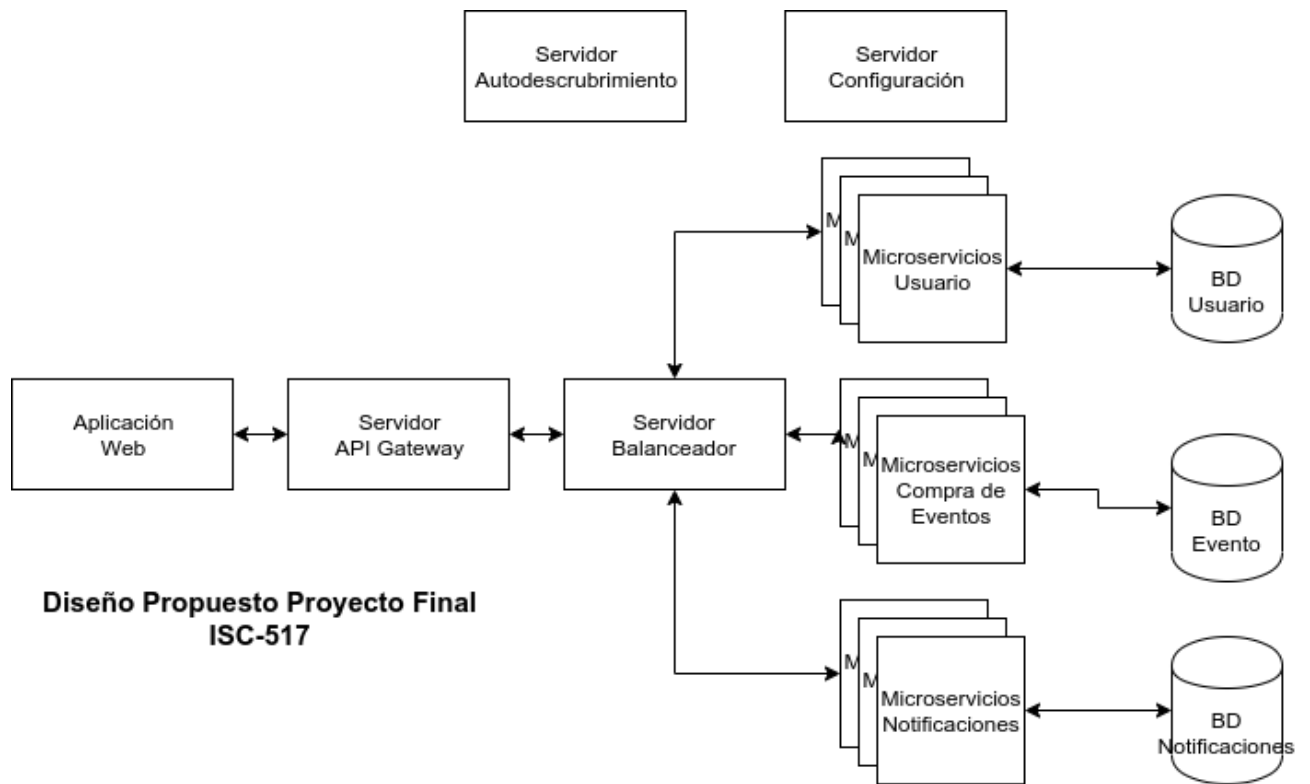
Dado el escenario planteado, la empresa está enfocada en una aplicación que trabaje 24 horas al día, los 7 días de la semana, para ellos nos solicitan los siguientes puntos:

1. Realizar una aplicación bajo la tecnología Java para el backend, utilizando un framework para el desarrollo. En la vista pueden utilizar cualquier framework JS o MVC con renderizado (Justificar la elección para ambos casos).
2. La aplicación a desarrollar debe implementar una arquitectura basada en microservicios y alta disponibilidad, propuesta en la imagen #1, donde cada uno de los componentes macros de los procesos de la aplicación trabajen de forma integrada e implemente una API (sincrónica o asíncrona) para la comunicación entre los bloques del sistema. (Explicar el diseño).
3. La aplicación desarrollada será accedida por el cliente vía web, el cual creará un usuario administrativo de forma automática. La aplicación permitirá la creación de usuarios del tipo cliente, vía registro directo de los usuarios que deseen comprar un paquete, una vez creado el usuario, el sistema enviará un correo electrónico al cliente con la información de acceso y enlace del sistema.
4. El cliente tendrá acceso a seleccionar productos disponibles en el sistema según el plan de paquetes (no es necesario hacer CRUD de productos para simplificar el proyecto), una vez seleccionados por el usuario, simulará un pago mediante tarjeta de crédito utilizar PayPal. Una vez completado el proceso de pago, retornará a la

aplicación donde visualizará un resumen de la transacción realizada de la factura generada en la transacción y enviando un correo transaccional al cliente con dicho resumen.

5. Una vez realizando el proceso de compra del paquete seleccionado, debe enviar un correo electrónico a los usuarios que están habilitados como empleados, indicando que es necesario asignarse un trabajo.
6. El sistema debe implementar el CRUD de los usuarios y perfiles del sistema, para indicar cuales usuarios son clientes y cuales son empleados del sistema.
7. El sistema debe permitir a los usuarios con privilegio poder visualizar todas las compras realizadas por todos los usuarios del tipo cliente.
8. El sistema debe permitir mostrar a los usuarios todas las compras realizadas.
9. El sistema debe implementar una vista que contenga gráficos de las compras realizadas, solicitudes pendientes y realizados del día actual.
10. **Se debe cuidar la presentación del sistema.**
11. El sistema debe ser seguro contra los accesos no autorizados, implementar framework de seguridad.
12. Es necesario que toda el diseño sea manejado en una arquitectura basada en microservicios.
13. Los microservicios que representan las funcionalidades tienen distribuir las peticiones entre tres (3) instancias mínimo.
14. Debe ser implementado un servicio de auto-descubrimiento y configuración centralizada.
15. Pueden utilizar la tecnología de Spring Cloud.
16. La base de datos será H2 modo servidor o Mysql individual para cada servicio.
17. **El proyecto debe utilizar Docker y correr vía Docker Compose versión 3.**

## Esquema propuesto de arquitectura basada en Microservicios:



*Ilustración 1: Diagrama de Aplicación*

### Notas:

- El servidor API Gateway y Balanceador pueden estar en un único proyecto en el caso de utilicen Spring Cloud Zuul con Ribbon.
- Para el manejo de la memoria pueden ejecutar su proyecto desde la [Play With Docker](#).