

# Neuss Producciones Ltda.

## **Integrantes:**

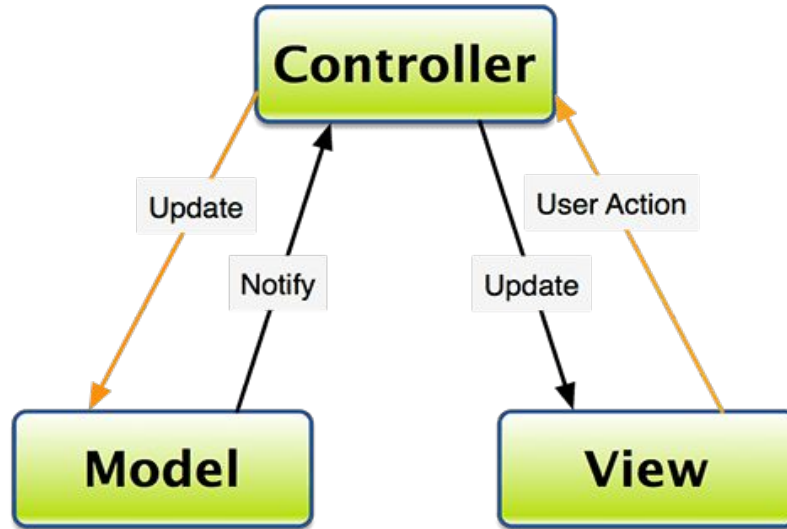
- Andrés Gutiérrez Salas
- Miguel Nahuelpán Aliante
- Esteban Tejeda Webar

# Propósito del proyecto

El propósito de este proyecto es buscar desarrollar una aplicación capaz de conectar con un público dispuesto a consumir platillos elaborados por la empresa utilizando un enfoque Dark Kitchen, ofreciéndoles a ellos un medio para poder cotizar y seleccionar las distintas ofertas que se ofrezcan, requieren un proyecto que consiste en ayudar y facilitar la comunicación proveedor-cliente.

# Arquitectura de Software

Luego de distintas reuniones examinando cada tipo de arquitectura presentada en clases, se tomó la decisión de utilizar el Modelo-Vista-Controlador (MVC)



# Fundamentos que sustentan la decisión

Se examinó a detalle cada arquitectura vista en clases:

- Event-driven architecture: Se descartó esta opción ya que al estar basado en eventos, tiene mayores usos en los eventos generados con los navegadores o en la comunicación con algún tipo de hardware. Como el software a desarrollar no requiere una fuerte interacción en el navegador y en los eventos que ejecute el usuario, entonces se descarta.
- Service oriented architecture: Se descartó esta opción ya que busca la integración de distintas bases de datos en un solo sistema. Dado que en este sistema no se requieren consultas de distintas bases de datos entonces se descarta.

# Fundamentos que sustentan la decisión

- Space-based architecture: \*FALTA\*
- Layered architecture: La arquitectura basada en capas permite segmentar un proyecto en varias capas, cada una de forma independiente. Esto podría hacer que el desarrollo del proyecto sea más tedioso y largo para los requisitos solicitados. Por lo tanto se descartó.

# Fundamentos que sustentan la decisión

Por último, quedamos entre dos arquitecturas: microservice y MVC.

Esta fue la decisión que más nos costó, ya que ambas poseen bastantes similitudes.

Finalmente se descartó el uso del microservicio ya que en la interfaz, cada microservicio define una especialización, como también cada uno posee una base de datos independiente a las anteriores. Esto puede llegar a ser desaprovechado por el hecho de que en el proyecto no se requiere tanta segmentación en cuanto a una especialización, como también el hecho de que no es necesario el uso de muchas bases de datos.

# Enfoques de reutilización

- Los **frameworks** son un enfoque efectivo en la reutilización de alto costo para introducirse en el desarrollo de software, por lo que esta opción es descartada.
- **Sistemas de solución COTS**, es una aplicación genérica de un solo proveedor que se configura de acuerdo al cliente. Como el cliente no configura la aplicación se descarta este enfoque.
- **Sistemas COTS integrados**, incluyen 2 o más productos (sistemas), por lo que se descarta, ya que solo realizaremos un sistema.
- **Líneas de productos de software**, se le entiende como un conjunto de sistemas de mayor o menor complejidad, que se seccionan para poder desarrollar y crecer.

# Enfoque de reutilización utilizado en el proyecto: Líneas de productos software

Identificando las líneas de productos tenemos las siguientes:

- Línea de administración
- Línea de usuario

beneficios: al crear líneas de productos podemos simplificar el tiempo y costo de un proyecto, reutilizar componentes en común y generar componentes particulares de cada línea.

El desafío de esta propuesta es cómo lograr gestionar los componentes de cada línea.