Reunión 02 de agosto

# Dudas

* Motivación: ¿tendría que explicar a grandes rasgos que es una Arq. Monolítica y una basada en microservicios? Porque después se nombra en los objetivos. Los microservicios se explican más adelante también.
  + Experimentar xq ya hay información. Hacerlo de primera mano y compararlas de primera mano.
  + Auge en el interés por los microservicios. Tecnología que surge.
* Revisar los objetivos.
  + ¿Qué conclusiones se pueden obtener? (1 y 2)
  + ¿En lugar de escalabilidad, podría decir elasticidad o disponibilidad?
  + Incluir algo de programación o persistencia políglota.
* Estado del arte.
  + ¿Dividir en más capítulos?
  + En análisis, ¿explicar lo que es la escalabilidad y la elasticidad?
  + En implementación, incluir:
    - Ley de Conway. Si MICROSERVICIOS
    - Microservicios en la UI: hago yo las conclusiones en las reflexiones.
  + En Despliegue, ¿dejar Kubernetes y Docker? Están por ahora por considerarlas más independientes de plataforma.
  + Secciones de:
    - Comparación con otras arquitecturas: no
    - Crítica al estado del arte. Si. Tecnologías y tfg.
    - Propuesta. Decir la tecnología q empleo.
* Análisis del sistema:
  + Con descripción, CUs, y modelo de dominio. ¿Demasiado?
  + Plan de trabajo: Proceso de desarrollo.

tecnología utilizada: estado del arte.

* + - No encajan con el título del capítulo, pero es común a ambas soluciones.
* Solución monolítica:
  + ¿Mucho detalle? ¿Mucho código? ESTÁ OK
* Solución basada en microservicios:
  + Centrada sobretodo en las diferencias: ESTO DECIRLO EN PROCESO DE DESARROLLO
    - Uso de diferentes arquitecutras.
    - Uso de otro lenguaje de programación.
    - Uso de otra BD.
    - Separación en soluciones y BDs diferentes.
* Evaluación de las soluciones:
  + Mantenimiento: TEORICO
    - Mantenimiento correctivo: buscar un defecto gordo y ver cuánto tengo que cambiar.
    - Nueva relación entre servicios distintos.
    - Migración de infraestructura (AWS).
      * Se puede abordar de forma progresiva solo con microservicios.
  + Escalabilidad: mejor uso de recursos.
  + USAR RNF DE LA ISO. NO HABLAR DE ESCALABILIDAD.
  + EN OBJETIVOS, INCLUIR MÁS RNF TALES COMO 3.
    - En Azure, se puede escalar todo. (Service APP vs AKS)
    - Comparar tamaño de imágenes.
    - Kubernetes solo asegura que hay en todo momento X instancias. Es necesario investigar más para asegurar que escala con la demanda.
    - ¿Monolítica?
  + Tolerancia a fallos:
    - Kubernetes nos garantiza que reacciona a errores porque siempre tiene esas X instancias.
    - Monolítica.
* Glosario: ¿solo términos o también definiciones?
* Apéndices:
  + Tablas que recapitule ventajas e inconvenientes (más visual)
    - En la evaluación
  + Descripción del prototipo desarrollado.
  + Despliegue con Kubernetes y Azure.
  + (OPCIONAL) Crear cliente con NSwag. Esto en lugar de ser un apéndice, podría resumirse e incluirse en la sección (**Generación de la capa de proxy**).

# Trabajo pendiente

* Capítulos 5 (solución basada en microservicios), capítulo 6 (evaluación) y capítulo 7 (conclusiones). Los últimos dos a la espera de resolver dudas.
* Resumen, resum, abstract.
* Corregir motivación.
* Crear un mejor hilo conductor en el estado del arte.

Ley de Conway.

Spotify organización de equipos.

BACK FRONT problemas en el ritmo de desarrollo.

Separar estado del arte en 2.

Teoria: Introducción a los microservicos.

Tecnología: estado del arte.

Alternativas tecnológicas: AZURE, AWS, CONTENEDORES, ORQUESTADORES.

Critica al estado del arte: muchas tecnologías, aparición de nuevas continua, que da igual elegir tecnología.

Propuesta: decir la tecnología que esxoges. Decir que lo hago en la empresa. También decirlo en la motivación, profundizar en xq.

Especificación de requisitos del caso de estudio.

Queremos probar versatilidad. Una situación mas o menos típica.

Tecnologia utilizada a propuesta.

Diagrama de casos de uso.

Escalabilidad: buen rendimiento con cierta carga.

Utilización de recursos: USAR SOLO los recursos que necesitas.

Recuperación de fallo.

Rendimiento, utilización de recursos y capacidad de recuperación.

Elasticidad: UTILIZACION DE RECURSOS

Proceso de desarrollo en lugar de plan de trabajo.

Plan de trabajo. Todo lo que es el trabajo en un cronograma. Anecdótico.

Indicar trello.

Referencia **o arquitectura que empleamos en la empresa**.

En el proceso de desarrollo, explicar que en el monolítico nos centramos en la tecnlogia e implementación y en la de microservicios nos vamos a centrar en el diseño.

Consideraciones distintas.

Extension del plan de trabajo y organizacipn caps siguientes.

Resaltar que vamos a emplear tecnologías realistas en términos de exigencia, calidad. En proceso de desarrollo.

Motivacion: app lo más realista posible.

Resaltarlo también a nivel de conclusiones.

EN EVALUACIÓN, HABLAR DE OTRAS CONFIGURACIONES DISTNTAS: NOSOTROS HEMOS PROBADO UNA. MISMA BD, OTROS LENGUAJES, DISTINTAS MÁQUINAS.

LA PARTE DE IU EN EVALUACIÓN.

MÁS ALLA DE ESCALABILIDAD Y UTILIZACION DE RECURSOS.

NORMALMENTE NO SE CONSTRUYEN DOS APPS.

APLICACIONES LEGADAS. MIGRACIÓN PARCIAL. Solución para la refactorización.

Hasta el día 13.

En la primera semana de septiembre, otra revisión.

WORD.