

ARQUITECTURAS DE SOFTWARE 2023-2

Escuela Colombiana de Ingeniería “Julio Garavito”

Proyecto Sprint 2

Fecha de inicio: Noviembre 27 de 2023.

Fecha de finalización: Diciembre 11 de 2023.

Requerimientos de entrega para el Sprint Review:

El proyecto debe seguir los elementos de gestión de SCRUM. Una vez se haya definido qué historias del BackLog del producto irán al BackLog del Sprint, el equipo debe hacer una planificación, y mantener un control de tiempos que permitan generar continuamente un Burndown Chart. Esto lo deben realizar en la herramienta destinada para la gestión de las actividades, Azure DevOps.

Es importante recordar que todas las HU debe estar en DONE cumpliendo con todos los criterios de aceptación, el punto principal es demostrar el cumplimiento de la promesa de valor de producto, la cual está definida desde el Sprint Concept en torno a una solución en tiempo real.

Para el segundo Sprint del proyecto, se espera que el Developer Team se enfoque en cumplir los siguientes Atributos de calidad (Requerimiento no funcionales) definidos con el cliente:

Atributo de calidad Disponibilidad:

- ☐ Se debe plantear una arquitectura que permita escalar horizontalmente el Back-End de la aplicación y hacer una implementación de concepto de dicha arquitectura, no deberían estar todos los componentes en el mismo servidor, para esta funcionalidad debe especificar un escenario de calidad que refleje el atributo de Disponibilidad – Escalabilidad del sistema.
- ☐ Se debe agregar como restricción tecnológica el uso de la plataforma Azure para la ejecución del proyecto, se puede apoyar en máquinas virtuales o web apps.
- ☐ Hacer una implementación de dicha arquitectura, para esto debe existir una separación física de los componentes, lo más distribuida posible. Es decir, montar todos los componentes de la arquitectura en un mismo servidor perjudicará la evaluación.

Atributo de calidad Seguridad:

- ☐ Se deben especificar dos escenarios de calidad asociado al atributo de Seguridad, el equipo es libre de elegir cual escenario implementar.
- ☐ Los escenarios de calidad deben ser aprobado por el profesor.
- ☐ Se debe actualizar la arquitectura para validar que cumple con los escenarios anteriormente descrito.

- Se debe realizar la implementación que solucione los escenarios anteriormente descritos.
- Se recomienda revisar las tácticas de seguridad especificadas en el libro “Software Architecture in Practice” de Len Bass, Paul Clements y Rick Kazman.

Atributo de calidad Mantenibilidad:

- Se debe especificar un escenario de calidad asociado al atributo de Mantenibilidad, en el cual se especifique que la aplicación va a ser sometida a una revisión de código estático, esto se debe incluir dentro del ciclo de integración continua, este proceso en DevOps es conocido como Inspección continua, en el siguiente enlace puede encontrar información adicional al respecto:
 - <https://www.devopsschool.com/blog/what-is-continuous-inspection/>
- Se debe implementar dentro del ciclo de Integración Continua un paso para ejecutar el escaneo del código fuente por medio de SonarQube, para esto es necesario que investigue la forma en que su herramienta de IC realiza la integración con SonarQube o SonarCloud.
- **Bono:** Si logra que las todas las métricas de SonarQube se encuentren en estado “A” (Verde) tendrá una bonificación de 5 décimas sobre la nota del proyecto, se debe tener una cobertura de pruebas unitarias sobre el 40% de cobertura como mínimo.

Documento de arquitectura:

El documento de arquitectura debe incluir una sección de Atributos de Calidad, donde se indiquen ESCENARIOS para los atributos descritos anteriormente. Todos los escenarios de calidad deben ser consistentes con la arquitectura planteada, es decir, se debe evidenciar que los escenarios tienen algún componente dentro de la arquitectura que logra asegurarlo.

Debe diseñar mínimo 5 diagramas UML (Ej: Diagrama de contexto, Diagrama de componentes, Diagrama E-R, diagrama clases, diagrama de secuencia, diagrama de despliegue, diagrama de actividades) para la arquitectura propuesta (Debe tener en cuenta que los diagramas en ocasiones no son suficiente para representar toda la información esperada, utilice los recursos que considere necesario para que la arquitectura sea capaz de representar SOLA toda la información esperada).

Adicionalmente debe contar con los demás elementos claves para la creación de una arquitectura (Drivers Arquitectónicos): una referencia a los requerimientos funcionales, restricciones y supuestos arquitectónicos.

Este documento de arquitectura debe estar escrito sobre la wiki de Azure DevOps o en un documento Word.

Evaluación final:

Recuerde que no se aceptarán proyectos que no vengan respaldados por un sistema de control de versiones (historial GIT) que muestre el trabajo realizado por cada integrante. Aquellos que cumplan con este requisito, tendrán como evaluación:

Calificación del Proyecto:

- Implementación de la solución para el atributo de calidad de disponibilidad, demostración de la escalabilidad: 20%.
- Implementación de la solución para el atributo de calidad de seguridad, demostración de la escalabilidad: 20%.
- Implementación de la solución para el atributo de calidad de mantenibilidad, demostración de la escalabilidad: 20%.
- Documento de arquitectura con los componentes esperados: 20%
- Robustez y eficiencia: consideraciones en el manejo de la concurrencia, condiciones de carrera, eficiencia de las operaciones, etc. 10%.
- Correcta ejecución del Azure DevOps: Se va a validar que los componentes Scrum que están dentro de Azure DevOps se hayan utilizado de forma correcta (Burndown Chart, HU, Features, etc.) 10%
- Contribución individual al proyecto. Este ítem afectará la evaluación de los anteriores. Es decir, si hubo poca o ninguna contribución, la evaluación individual para el proyecto será escalada en la misma proporción.