

Proyecto: **Kolb**

Guatemala, Guatemala

11 de octubre de 2,021

Jefe de desarrollo Donald Leiva

Historial de revisiones:

Encargado	Fecha	Descripción de revisión	Versión
Grupo #2	08/10/2021	Pruebas unitarias	v.1.0.0
Grupo #2	08/10/2021	Pruebas de integración	v.1.0.0
Grupo #2	11/10/2021	Pruebas de estrés	v.1.0.0

Pruebas realizadas:

- Pruebas unitarias
- Pruebas de integración
- Pruebas de estrés

Resultados de Métricas

- Métricas de requerimiento:

Métricas de Requerimiento		Recomendaciones
Ambigüedad	En las características del sistema se define que la aplicación debe ser fácil de usar para el usuario. ¿Qué es fácil para el usuario? En la parte 3.1.1.6 se especifica que los resultados deben ser mostrados para el usuario de manera estética y agradable, sin embargo no se adjunta ninguna maqueta o especificación de qué se considera estético y agradable.	Se recomienda generar un documento que lleve registro de los cambios que se realicen en los requerimientos para poder medir así la volatilidad de este. Además, se recomienda separar la capa de negocio por medio de una API aislada con tal de poder escalar el proyecto en caso de crecimiento o cambios de
Volatilidad	No se puede medir la volatilidad dado que no hay un historial de cambios de dichos documentos	

Comprensibilidad	El proyecto no es escalable al no separar la capa de negocio con la capa de interfaz de usuario. Esto hace dependiente de manera interna al proyecto y no poder ser reutilizado en otros lados o en diferentes plataformas en caso exista problemas de compatibilidad.	requerimientos. A partir de esto se podrá generar un diagrama que amplíe el funcionamiento y lógica general del sistema. Por otra parte, se podrían generar diseño de Mock Ups para poder tener una idea visual y no tener que responsabilizar al equipo de desarrollo con un diseño.
Claridad en modelo	El modelo actualmente se ve diagramado de forma general, en la cual no se puede ver cómo funciona la conexión entre la capa de negocio y la capa de datos.	

- Métricas de diseño:

Métricas de Diseño		Recomendaciones
Integridad arquitectónica	La conexión directa entre el modelo a la base de datos no es de buena práctica, ya que para eso existen capas (una API) como capa intermedia.	-Poder aplicar modelos de diseño dado que en este momento lo que se esta realizando es una aplicación pero que encapsula todo y su escalabilidad es limitada y requiera de más trabajo a la hora de utilizar otros dispositivos o hacer un cambio de framework web
Complejidad de componentes	No se puede realizar una medición de la complejidad al no tener enlistado los componentes.	
Complejidad de interfaz	No se puede medir dado que no hay diseños de interfaz con lo cual esto queda muy ambiguo.	
Patrones	El modelo MVC busca separar el contacto con alguna base de datos dado que eso debería ser utilizado por alguna API para poder realizar escalabilidad del proyecto dado que esto nos obliga a tener que separar el backend cuando podemos separarlo de una sola vez.	

- Métricas de código:

Métricas de código		Recomendaciones
Complejidad	Todo el sistema se encuentra en un mismo proyecto con lo cual la complejidad del mismo crece porque se adapta únicamente a lo que se esta realizando y es difícil de escalarlo.	Al tener una versión inicial del proyecto, sería importante tener la documentación para poder medir aplicar las métricas. Además, parece que el código
Comprensibilidad	El readme es comprensible y bien documentado. La	

	base que tomará el código es entendible, aunque sería mejor mostrar código funcional.	actual no está terminado, por lo que no podemos realizar mediciones sin funcionalidades completas. Se recomienda tratar las vulnerabilidades con los datos sin cifrar.
Reusabilidad	No existe historial de versionamiento de código (creación de ramas)	
Documentación	El archivo Readme.md sí proporciona una buena guía de utilización del sistema. El código se encuentra perfectamente documentado en cada uno de los métodos.	

- Métricas del sistema:

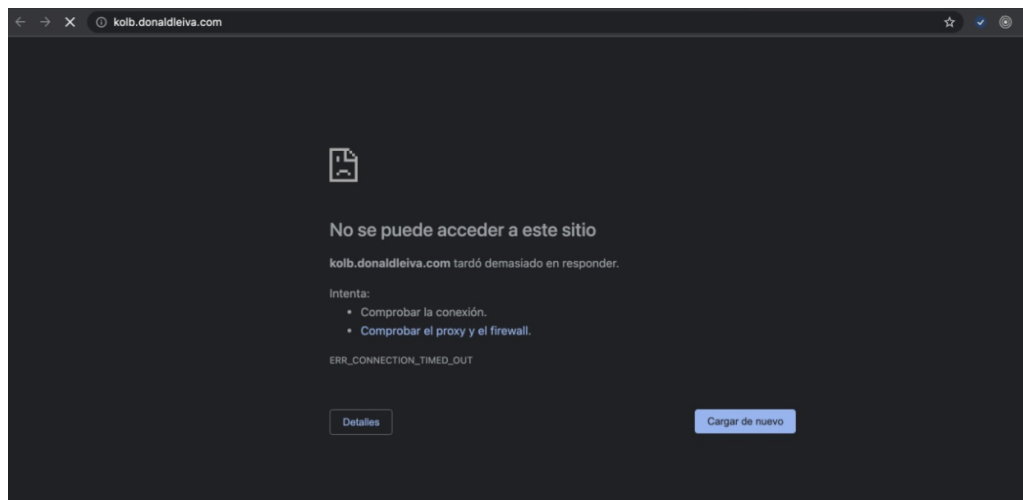
Métricas de sistema		Recomendaciones
Subconjunto de cambios	Se puede, incrementar la funcionalidad, del software.	Se recomienda visualizar el tutorial de forma obligatoria ya que como usuario común, uno ignora el popup del tutorial mostrado y prefiere realizarlo de forma rápida aunque la mayoría de veces incorrecta.
Integridad conceptual	No lo cumple, puesto que no existe una visión clara del sistema	
Sujeto a pruebas	Sí lo cumple, puesto que el equipo tuvo acceso y pudo realizar pruebas de sistemas	
Funcionalidad	Sí lo cumple, ya que se pudo realizar las tareas necesarias.	
Disponibilidad	Si lo cumple dado que este tiene un método de recuperación a la hora de caer ha de ser dado por el host	
Ser construido	Sí lo cumple, puesto que la version evaluada se realizó durante el tiempo disponible de entrega y funciona de forma correcta aunque poco intuitivo.	
Reutilización	Posiblemente el sistema de preguntas y respuestas puede ser utilizado para otros tipos de test psicológicos.	
Confiabilidad	Si funciona y esta disponible la mayoría de tiempo. Deja de funcionar cuando hay concurrencia de mínimo 4728 usuarios en la página.	

Variabilidad	Es complicado que se adapte a una nueva arquitectura, ya que esta implementado al sistema. Y si se desea cambiar la arquitectura se debe de cambiar bastante código.	
Portabilidad	Es portable en caso de cambiar lo de host o servidor. Pero para cambiarlo de sistema es muy poco portable.	
Integración	Es complejo integrar otros sistemas ya que el software se diseñó y construyó exclusivamente para funcionar con este test.	
Rendimiento	Usuarios/minuto: 4,727.975. Tiene una cantidad de resistencia a usuarios alta por el tamaño del proyecto y su uso.	
Modificable	Sí lo cumple, puesto que la estructura del proyecto está bien organizada y los requerimientos se pueden implementar sin mayor problema	
Usabilidad	No lo cumple, puesto que la aplicación es poco intuitiva y para usuarios debajo del nivel en usos de tecnologías es difícil de entenderlo y operarlo.	
Seguridad	El documento del proyecto no se puede validar la métrica de seguridad ya que no se especifica si este necesita de alguna identificación de usuario o cualquier persona puede acceder.	

Evidencias:

[Métricas - Hojas de cálculo de Google](#)

[JMeter](#)



✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1
✖ Failed to load resource: the server responded with a status of 422 ()	kolb/results:1

Resultado de pruebas:

Rechazar por errores de seguridad, escalabilidad y usabilidad.

Fecha de próxima entrega:

18 de octubre de 2,021

Autorizado por:



Eduardo Albarizaez



Marcos Calderón



José Díaz



Mario Velasquez



Boris Morán