

DISEÑO Y DESARROLLO APLICACIÓN EN JAVA QUE PERMITA REGISTRAR Y PRESENTAR LOS RESULTADOS DE LAS PRUEBAS DE CALIDAD EJECUTADAS POR EL EQUIPO DE QA.

Para el diseño y desarrollo de esta aplicación se usó el framework LPGL SMARTGWT versión 6, framework GWT, base de datos en Microsoft SQL Server 2017 Express, IDE Eclipse 2018-12 R y JUnit para desarrollar las pruebas. En la carpeta Resources se encuentra un video donde se detalla el funcionamiento.

BASE DE DATOS

En el diseño y desarrollo de la base de datos, se consideró:

- la creación de usuarios para tener un rol de trabajo (usuario QA y usuario administrador);
- la creación de aplicaciones, junto a sus respectivas versiones (nombradas a libertad).
- Cada prueba registrada se relaciona con el usuario que la ejecuto, la aplicación y numero de versión específica, resultado y descripción de la prueba, tiempo de inicio y tiempo de finalización y las métricas consideradas.

El diseño se ilustra en el modelo entidad relación de la imagen 1.

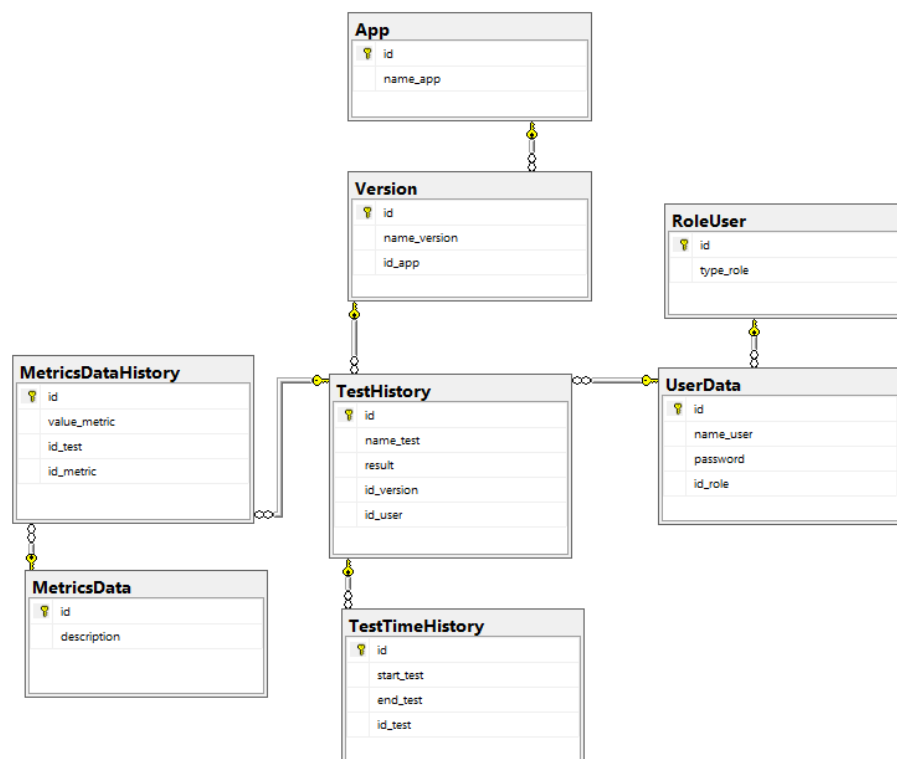
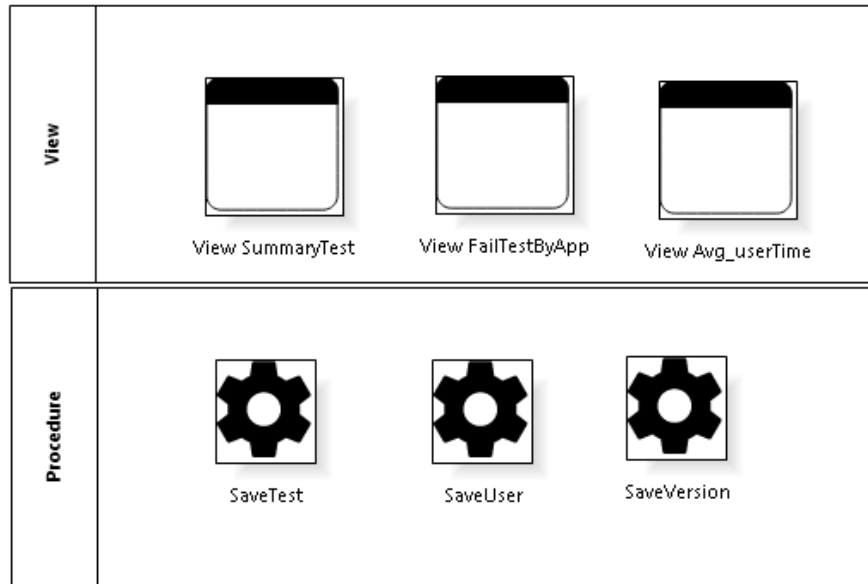


Imagen 1. Modelo entidad relación DB QA.

se construyeron vistas, con el fin de preparar y procesar la información de una manera más eficiente, evitando cargas lentas en la aplicación. Los procedimientos simplifican el código necesario para guardar los registros de las pruebas, guardar los usuarios y las versiones de cada aplicación.

- View SummaryTest = Histórico de todas las pruebas realizadas.
- View FailTestByApp = Porcentaje de pruebas fallidas según las métricas consideradas agrupadas por aplicación y versión.
- View Avg_userTime = promedio del tiempo que demora cada usuario en realizar una prueba.



Powered by
bizagi
Modeler

Imagen 2. View y Procedure DB QA.

MÉTRICAS

Las métricas registradas por el usuario son:

- ¿Qué tan satisfecho está con la facilidad de uso de esta aplicación?: El acceso a las funcionalidades de aplicación debe ser fácil e intuitivo para los usuarios. El rango de valoración es de 1 a 5, si es 3 o menor se considera fallida.
- ¿Qué tan exitosa fue la prueba?: Considerar si la prueba cumplió con los requerimientos exigidos. El rango de valoración es de 1 a 5, si es 3 o menor se considera fallida.
- ¿Cuántos fallos fueron detectados?: la cantidad de errores detectados en la prueba.

Métrica que se calculan con registro de pruebas.

- Cantidad de pruebas por aplicación y versión.
- Tiempo que se demora en realizar una prueba.
- Porcentaje de pruebas fallidas por usabilidad.

- Porcentaje de pruebas fallidas por no cumplir los requerimientos.
- Promedio en que tarda un usuario QA en ejecutar una prueba.

APLICACIÓN QA

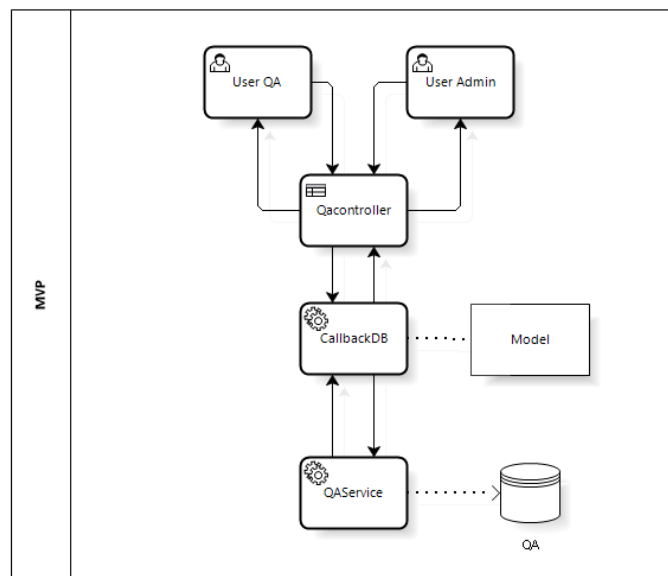
En el diseño de la aplicación se consideró la arquitectura MVP (ver imagen 3), teniendo en cuenta la documentación de SMARTGWT Y GWT. Su entada es por medio de un usuario y contraseña que verifica el rol del usuario.

Para el rol administrador la interfaz permite

- Crear aplicaciones.
- Crear versiones y asignarla a una aplicación específica.
- Crear usuarios y asignarles un rol.
- Una pestaña para visualizar el histórico de pruebas y sus métricas respectivas.
- Una pestaña para visualizar el porcentaje de pruebas fallidas según las métricas.
- Una pestaña para tener el tiempo promedio en realizar las pruebas por usuario.

Para el rol usuario QA

- Una ventana donde debe seleccionar una aplicación y una versión previamente creadas, iniciar la prueba, llenar el formulario y finalmente guardar.



Powered by
bizagi
Modeler

Imagen 3. Arquitectura MVP.