Entrega final de las 3 materias

Hember Güiza Cárdenas

Juan Miguel Castro Rojas

Docente:

Adán Beltrán Gómez

Docente del Área.

Fundación Universitaria Empresarial de la Cámara de Comercio de Bogotá

Escuela de Ingeniería

2023

# Problema y solución planteada

El proyecto propuesto se trata de un sistema de subsidios donde las personas pueden registrarse, aplicar a un subsidio y enterarse al momento de si serán aprobados o no para recibir el beneficio, así mismo este sistema va a administrar automáticamente los desembolsos y el proyecto al cual está asociado dicho subsidio.

La solución estará hecha usando una arquitectura orientada a eventos con dos clientes, uno web y uno de escritorio y una base de datos relacional en SQL Server.

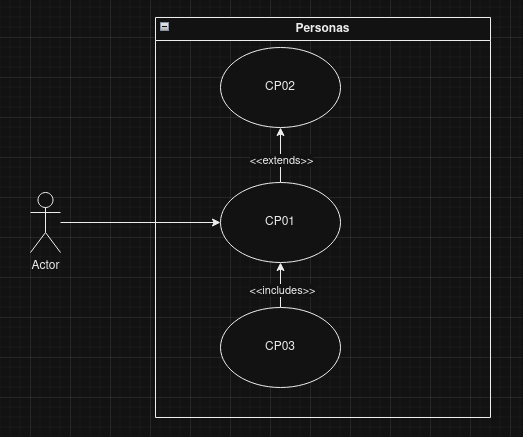
Construcción y Arquitectura de Software

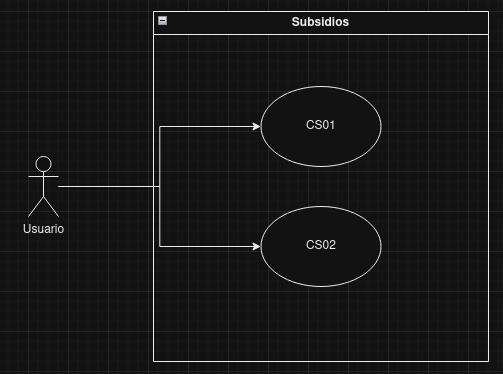
## Requerimientos funcionales y no funcionales acorde plantilla

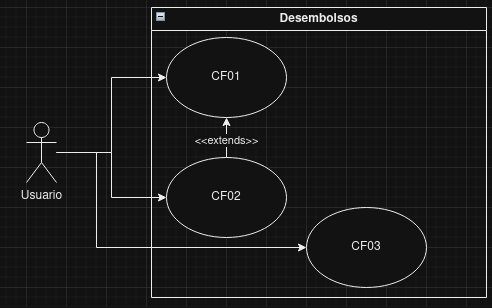
|  |  |  |  |
| --- | --- | --- | --- |
| **No funcionales** | | | |
| Id | Categoria | Descripción Requerimiento | Descripción CA |
| 1 | **Adecuación Funcional** | Cada campo de texto o de selección debe estar validado según la información que se solicite. | 1. al momento de ingresar un valor erróneo, no se podrá continuar con el proceso que se esté realizando y deberá señalarse el campo en donde se está cometiendo el error 2. al momento de continuar con el proceso cada campo deberá estar validado asertivamente, de lo contrario se deberá señalar el campo con información errónea y no se podrá continuar |
| 2 | **Adecuación Funcional** | El sistema debe dar al usuario información precisa según la normatividad para subsidio de vivienda en Colombia | la información dada por el sistema debe coincidir con la información solicitada y registrada en la plataforma por un asesor |
| 3 | **Adecuación Funcional** | el sistema debe mostrar mensaje de error al momento de diligenciar erróneamente la información en el login, signup, creación y modificación de subsidios e intentar ingresar. | al momento de ingresar erróneamente e intentar continuar con el proceso, se visualizará el mensaje de error con el respectivo mensaje |
| 4 | **Usabilidad** | La interfaz de usuario debe ser intuitiva y fácil de usar, especialmente para personas con diferentes niveles de habilidad tecnológica. | las gentes deben ser capaces de entender y navegar por la plataforma con facilidad y comodidad |
| 5 | **Seguridad** | El sistema debe mantener los principios de confidencialidad para el acceso a la información personal |  |
| 6 | **Seguridad** | el sistema debe ocultar las contraseñas introducidas | el sistema en los campos donde se ingrese o visualice contraseñas, deberá poner "•" en lugar de cada carácter |
| 7 | **Seguridad** | el sistema debe guardar las contraseñas en la base de datos bajo encriptación sha-256 | al momento de consultar en la base de datos la contraseña de las cuentas de los asesores estas deberán estar encriptadas |
| 8 | **Mantenibilidad** | Agregar nuevas funcionalidades o modificar funcionalidades al sistema debe ser alcanzable dentro de un rango de 1 a 2 semanas (mantener unidad cohesiva de SPRINTS). | Los cambios que pida el cliente o los errores que se presentan en producción se resuelven en menos de 1 a 2 semanas. |
| 9 | **Portabilidad** | El sistema debe ser capaz de desplegarse en plataformas web, así como también en desktop | 1. El sistema se despliega y funciona correctamente en navegadores web y en escritorio 2. La interfaz de usuario es responsiva y se adapta adecuadamente a diferentes tamaños de pantalla. 3. Todas las funciones principales del sistema están disponibles y son utilizables en los diferentes dispositivos mediante navegador web y escritorio. 4. Las acciones comunes, como completar formularios y enviar solicitudes, pueden realizarse sin dificultades significativas en los diferentes dispositivos mediante navegador web y escritorio. |
| **Funcionales** | | | |
| Id | Categoría | Descripción Requerimiento | Descripción CA |
| 10 | Funcionales | el sistema debe permitir hacer operaciones CRUD con personas | 1, al momento de realizar la eliminación aparecerá un mensaje que confirme la acción  2, al momento de hacer una acción de registro, modificación o eliminación de cuentas de usuario, este cambio debe aparecer en la tabla correspondiente |
| 11 | Funcionales | el sistema debe permitir hacer operaciones CRUD con aplicaciones | 1, al momento de hacer una acción de registro, modificación o eliminación de cuentas de usuario, este cambio debe aparecer en la tabla correspondiente |
| 12 | Funcionales | el sistema automáticamente se encarga del proceso de evaluación de aplicaciones de subsidio | 1. al momento de realizar la aplicación a un subsidio se visualizará en la lista de aplicaciones junto con el estado predeterminado "Pendiente" mientras el sistema calcula. 2. el sistema cambiara el estado de la aplicación al correspondiente según lo que haya calculado internamente usando los procedimientos que se le dieron |
| 13 | Funcionales | el sistema automáticamente ejecuta los desembolsos mientras la aplicación al subsidio este aprobada y aun no haya expirado. | Cada mes el sistema ejecutara lo necesario para hacer efectivo un desembolso. Cada vez que se ejecute un desembolso el sistema debe agregar una fila a la tabla correspondiente |

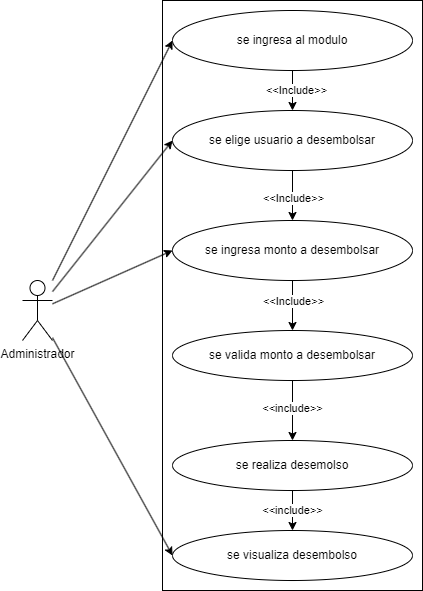
## Modelado de casos de uso

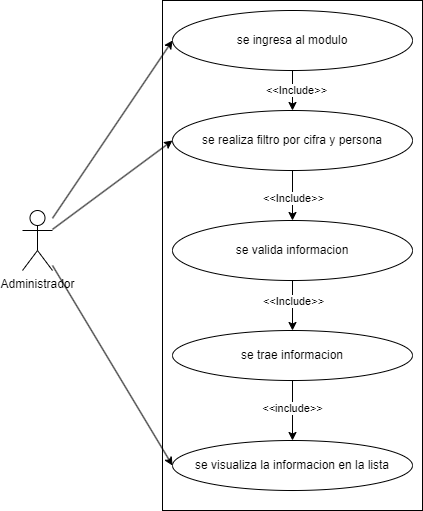
|  |  |  |  |
| --- | --- | --- | --- |
| **Nomenclatura** | **Modulo** | **Casos de uso** | **Responsable** |
| CP01 | Personas | Crear una cuenta | Ciudadano |
| CP02 | Personas | Ingresar a la cuenta | Ciudadano |
| CP03 | Personas | Solicitar eliminación de la cuenta y datos de la persona. | Ciudadano |
| CS01 | Subsidios | Aplicar a un subsidio | Ciudadano |
| CS02 | Subsidios | Evaluar aplicación de subsidio | Sistema |
| CF01 | Desembolsos | Ejecutar desembolso | Sistema |
| CF02 | Desembolsos | Finalizar subsidio | Sistema |
| CF03 | Desembolsos | Listar subsidios por cifra y persona | Ciudadano, Asesor. |

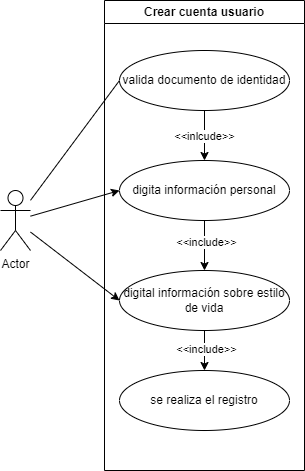


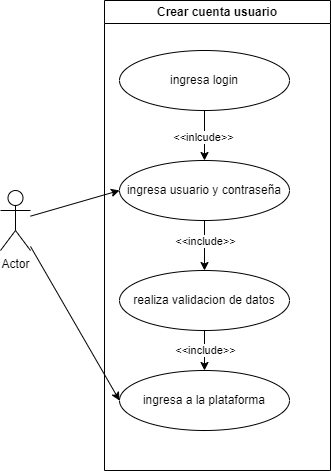


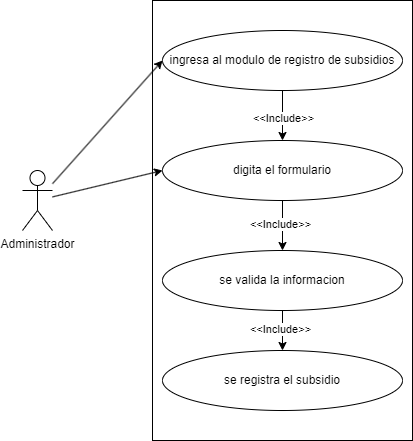


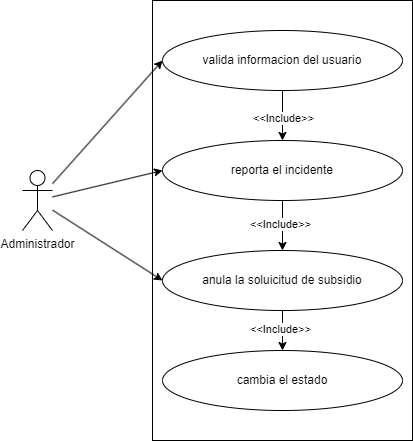


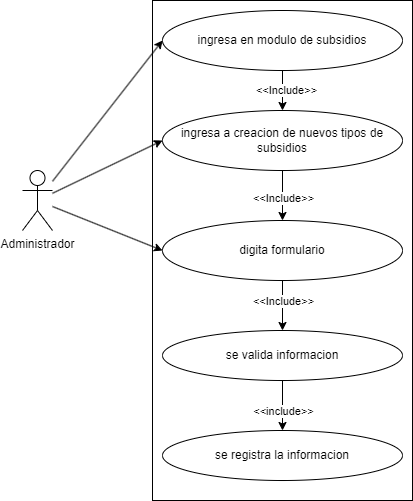


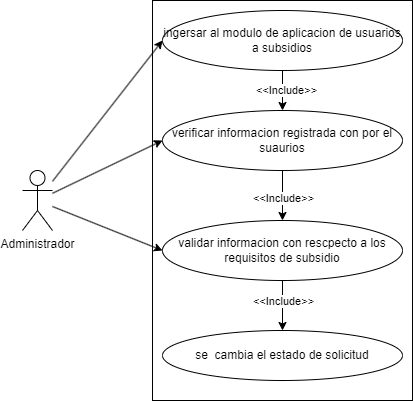




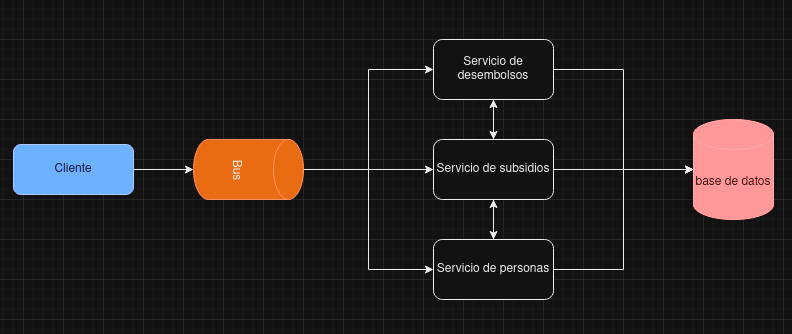








## Modelado de arquitectura



## Diseño Detallado de casos de uso

|  |  |  |  |
| --- | --- | --- | --- |
| **Información de Catalogación** | | | |
| **Proyecto** | *Sistema de subsidios* | | |
| **Autor** | *Hember Güiza Cárdenas* | | |
| **Versión** | *0.1* | **Estado de Desarrollo** | *Pendiente revisión* |
| **Definición del Caso de Uso** | | | |
| **Código** | *CS02* | | |
| **Nombre** | *Evaluar aplicación de subsidio* | | |
| **Objetivo** | *Actualizar el estado de una aplicación existente de ‘Pendiente’ a ‘Rechazado’ o ‘Aceptado’ según corresponda.* | | |
| **Descripción** | *Convertir un registro de la tabla de Aplicaciones en aplicación rechazada o aceptada dependiendo del tipo de subsidio.* | | |
| **Actores** | *Ciudadano, Sistema* | | |
| **Condiciones Necesarias** | *El ciudadano esta registrado ante el sistema.*  *El ciudadano ya creo previamente una aplicación a un subsidio particular* | | |
| **Escenario Principal** | *1 el cliente termino el proceso de diligenciar su aplicación a subsidio en particular.*  *2 el sistema busca la aplicación que se acaba de generar por ID*  *3 el sistema trae a memoria los datos del tipo de subsidio al cual se está aplicando.*  *4 el sistema ejecuta el procedimiento almacenado correspondiente según el tipo de subsidio.*  *5 el sistema compara el valor que se ha diligenciado en cada requisito con el valor que se supone debe tener para aprobar el requisito particular.*  *6 el sistema actualiza el estado de la aplicación a ‘Aprobado’ y pone la fecha actual a fecha inicio, mientras que pone fecha fin a 12 meses después de la fecha actual.* | | |
| **Escenario Alternativo** | *6a. Uno de los requisitos tiene un valor diligenciado diferente al que se supone debe tener, el sistema actualiza la aplicación a ‘Rechazado’. El caso de uso se termina.* | | |
| **Escenarios de Excepción** | *2a. El sistema informa al cliente de que la aplicación que intenta evaluar no existe, el caso de uso se termina.*  *3a El sistema informa al cliente que se está intentando evaluar un tipo de subsidio que no existe.* | | |
| **Condición de éxito** | *El registro correspondiente a la aplicación que el usuario creo ha cambiado de estado el cual será un valor diferente a ‘Pendiente’.* | | |
| **Cuestiones a resolver** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Información de Catalogación** | | | |
| **Proyecto** | *Sistema de subsidios* | | |
| **Autor** | *Hember Güiza Cárdenas* | | |
| **Versión** | *0.1* | **Estado de Desarrollo** | *Pendiente revisión* |
| **Definición del Caso de Uso** | | | |
| **Código** | *CF01* | | |
| **Nombre** | *Ejecutar desembolso* | | |
| **Objetivo** | *Enviar la cifra de dinero correspondiente a la aplicación de subsidio.* | | |
| **Descripción** | *Cada 30 días el sistema busca las aplicaciones que aún no han terminado y están activas para ejecutarles el desembolso.* | | |
| **Actores** | *Ciudadano, Sistema* | | |
| **Condiciones Necesarias** | *El ciudadano esta registrado ante el sistema.*  *El ciudadano ya creo previamente una aplicación a un subsidio particular* | | |
| **Escenario Principal** | *1 se completa un ciclo de 30 días*  *2 el sistema busca en base de datos aplicaciones a subsidio que estén estado ‘Aceptado’ y cuya fecha no sea inferior a la fecha actual.*  *3 el sistema itera, por cada resultado devuelto, envía la cifra correspondiente y genera un registro con la fecha actual y la cifra que envió.* | | |
| **Escenario Alternativo** | *3a el registro por el cual está iterando actualmente, 30 días en el futuro no cumple con la condición para poder recibir otro desembolso, el sistema actualiza dicha aplicación a estado ‘Finalizado’* | | |
| **Escenarios de Excepción** | *3a algún campo de la aplicación tiene datos nulos, omite el registro para pasar al siguiente.* | | |
| **Condición de éxito** | *Cada 30 días se ha agregado un registro a la tabla Desembolsos con la fecha actual y la cifra correspondiente.* | | |
| **Cuestiones a resolver** |  | | |

Database Administrator

Diseño normalizado

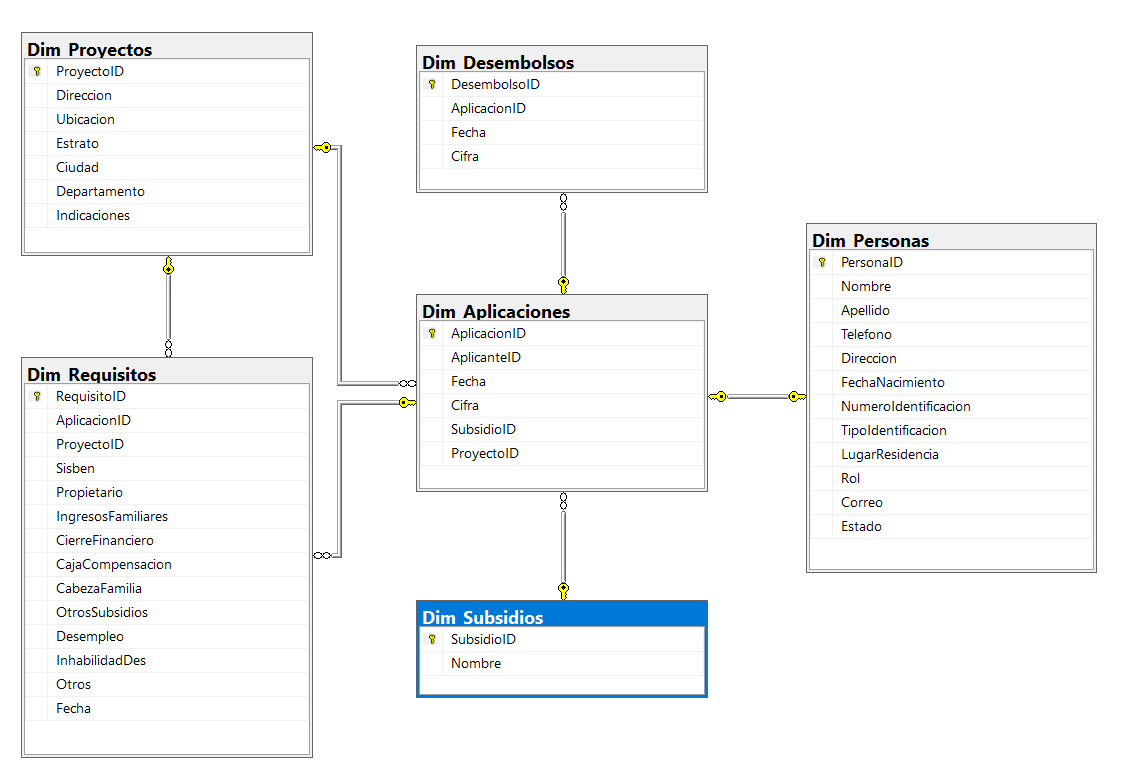


Para ver el SQL consultar el repositorio.

Triggers de auditoria

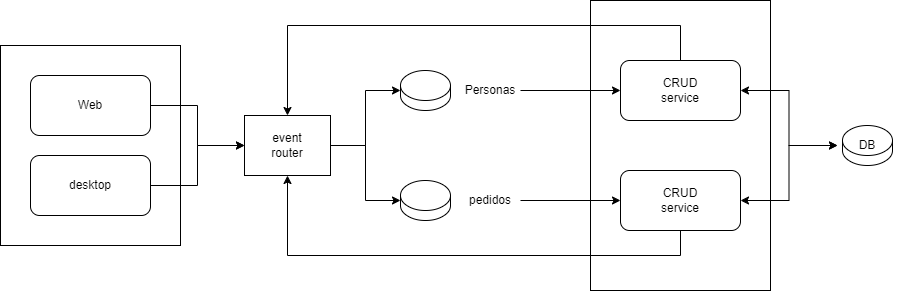
Ver archivo: <https://github.com/jucastro94/proyecto-subsidios/blob/main/DBA/SQL/triggersSQL.sql>

Diseño Data Warehouse para operaciones CRUD



Para ver el SQL consultar el repositorio.

Arquitectura



Aplicaciones de Escritorio

## Implementación de la arquitectura

Ver en el repositorio: <https://github.com/jucastro94>

## Documentación de casos de prueba

## Automatización de las pruebas unitarias

Ver en el repositorio: <https://github.com/jucastro94>