

TRABAJO PRÁCTICO DE SEGUIMIENTO

OBJETIVO

El siguiente trabajo práctico será utilizado como una herramienta de integración para el desarrollo de los temas propuestos por la cátedra sobre un dominio específico de aplicación, desde la creación de un modelo de datos hasta la construcción y prueba de un aplicación Windows (formularios) que permita gestionar los requerimientos funcionales del enunciado.

INTRODUCCIÓN

Un sistema de seguimiento de errores es una aplicación diseñada para ayudar a garantizar la calidad de software y asistir a programadores y otras personas involucradas en el desarrollo y uso de sistemas. El acrónimo BTS (Bug Tracking System) es el término usado generalmente para referirse a este tipo de sistemas.

Nuestro desafío será modelar una aplicación Windows sobre la plataforma .NET que permita implementar los procesos core desde el registro de errores hasta la emisión de reportes estadísticos por diferentes criterios.

A continuación se describen los procesos de sistema (casos de uso) que serán soportados por la aplicación junto con sus descripciones a trazo grueso, diagrama de clases de dominio, diagrama de CU y Modelo Entidad-Relación propuestos. Adicionalmente se presenta un diagrama de estados para la entidad Bug.

PROCESOS PRINCIPALES

A continuación se enumeran los casos de uso que serán soportados por la aplicación.

1. Registrar inicio de sesión.

Perfil: TODOS

Escenarios:

- El **usuario** de la aplicación ingresa sus credenciales (usuario y clave) y el sistema valida que exista en la base y que sus credenciales son correctas. Se habilita la pantalla principal de la aplicación mostrando un menú de opciones según su perfil junto con: [Nombre usuario + Perfil] en la parte superior de la interfaz.
- El **usuario** no existe o las credenciales son incorrectas. Se invalida el ingreso a la aplicación y se informa la situación.

2. Consultar errores reportados

Perfil: Tester, Desarrollador, Administrador (Consultor de Bugs).

Escenarios:

- El **Consultor de Bugs** selecciona el/los filtros de búsqueda (por producto, por prioridad, por criticidad, rango de fechas y estado). El sistema busca y muestra los resultados en una grilla ordenados por fecha de creación (por defecto se muestran los 20 errores más recientes). Además permite *Consultar el detalle* completo de los datos de error reportado.
- El sistema no encuentra resultados para la combinación de criterios solicitados. Se informa la situación y se limpian los criterios de consulta ingresados.

3. Consultar detalle de error reportado

Perfil: Tester, Desarrollador, Administrador (Consultor de Bugs).

Escenarios:

- El sistema muestra en modo lectura los datos completos de un error previamente seleccionado indicando: fecha de creación, fecha última modificación (formato dd/mm/yyyy hh:ss) y estado actual del error.

4. Actualizar usuarios

Perfil: Administrador.

Escenarios:

- El sistema muestra el listado de usuarios registrados y permite registrar y/o actualizar los usuarios de la aplicación (Se muestran por defecto los últimos 20 usuarios creados permitiendo consultar por nombre, por perfil o todos).
- El **Administrador** selecciona un usuario existente y actualiza exitosamente sus datos. Principalmente puede actualizar su nombre y/o clave, y su perfil o rol para el sistema.
- El **Administrador** selecciona la opción *Nuevo*. El sistema permite ingresar los datos completos, validando datos requeridos. A su vez se valida que no exista otro usuario con el mismo nombre. Los datos son validados y el registro se realiza con éxito. Al crearse el usuario queda automáticamente habilitado para comenzar a operar con el sistema.
- Tanto para el registro como para la actualización de usuarios, si los datos no son completos y/o correctos se informa la situación y se cancela la funcionalidad.
- El **Administrador** selecciona la opción *Habilitar/Deshabilitar*. El sistema permite actualizar el estado del usuario. Solo pueden iniciar sesión los usuarios habilitados.

5. Actualizar error reportado

Perfil: Tester, Administrador, Desarrollador.

Escenarios:

- El sistema muestra el listado de errores reportados. Se llama al proceso: “*Consultar errores reportados*”
- El **Tester** selecciona un error existente y actualiza exitosamente sus datos. Principalmente puede actualizar su prioridad, criticidad, asignado a y estado. Cada vez que un error reportado por **Tester se asigna a un programador**, se llama al proceso: “Asignar error a desarrollador”
- El **Tester** selecciona la opción *Nuevo*. El sistema permite ingresar los datos completos de un error reportado, validando datos requeridos. Los datos son validados y el registro se realiza con éxito.
- Tanto para el registro como para la actualización de errores, si los datos no son completos y/o correctos se informa la situación y se cancela la funcionalidad.

- Cada vez que se crea o actualiza un error el sistema genera un historial de bug que permite contar con un seguimiento de todas las intervenciones que sufrió el bug desde su creación hasta su cierre.

Cada historial registra: quién es responsable del cambio, en qué momento, en qué estado queda y eventualmente a quién queda asignado (Ver diagrama de estados anexo).

6. Asignar error a desarrollador

Perfil: Tester, Administrador.

Escenarios:

- El sistema permite asignar un error a un programador para su posterior solución. Cada vez que un **Tester** asigna un error a un usuario con perfil **programador**, el sistema registra un historial de asignación permitiendo conocer en todo momento: Cuándo, quién y en qué estado se encuentra dicho error.
- Si el error se encuentra en estado: **Cerrado**, se informa la situación y se cancela la funcionalidad.

7. Generar reporte de errores por tipo

Perfil: Responsable de Reportes, Administrador (Consultor de Reportes).

Escenarios:

- El **Consultor de Reportes** selecciona el/los filtros de búsqueda (por producto, por prioridad, por criticidad, rango de fechas y estado). El sistema genera un reporte contabilizando los errores consultados según los filtros indicados. Se muestran porcentajes de errores por estado y producto agrupados por Tester responsable.

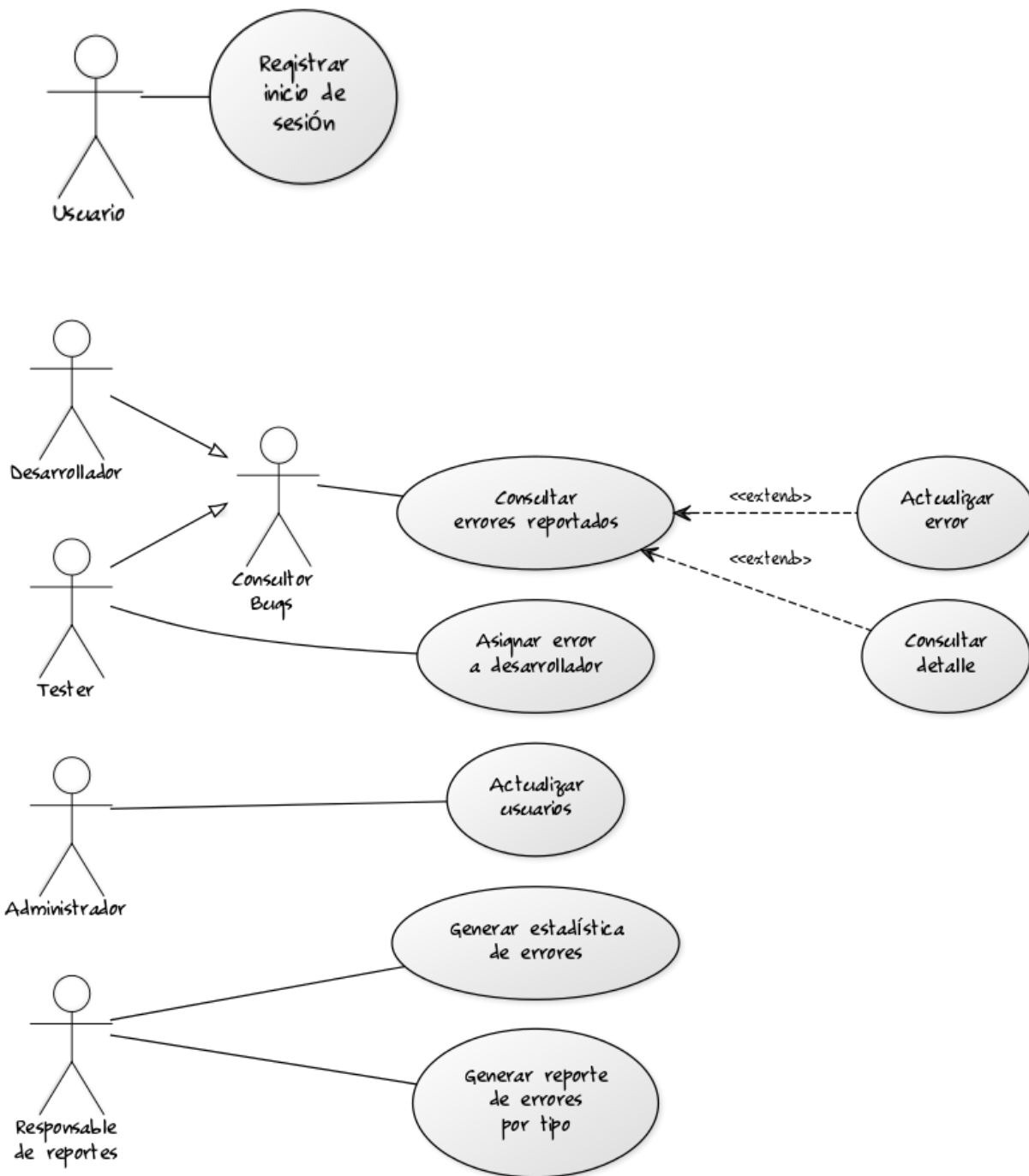
8. Generar estadística de errores detectados

Perfil: Responsable de Reportes, Administrador (Consultor de Reportes).

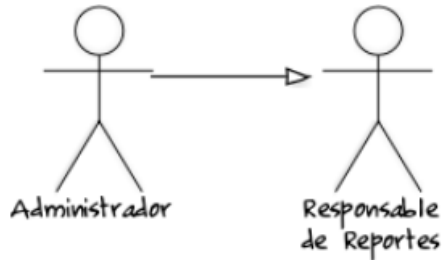
Escenarios:

- El **Consultor de Reportes** selecciona periodo de consulta. El sistema genera un reporte contabilizando los errores consultados para el periodo ingresado. Se muestra un gráfico de barras con los totales de errores reportados sin solucionar, contra cerrados, agrupados por producto.

DIAGRAMA DE CASOS DE USO



Generalización de actores



NOTA:

Adicionalmente, los actores: Consultor de Bugs y Responsable de Reportes son generalizados en **Usuario**.



DIAGRAMA DE CLASES:

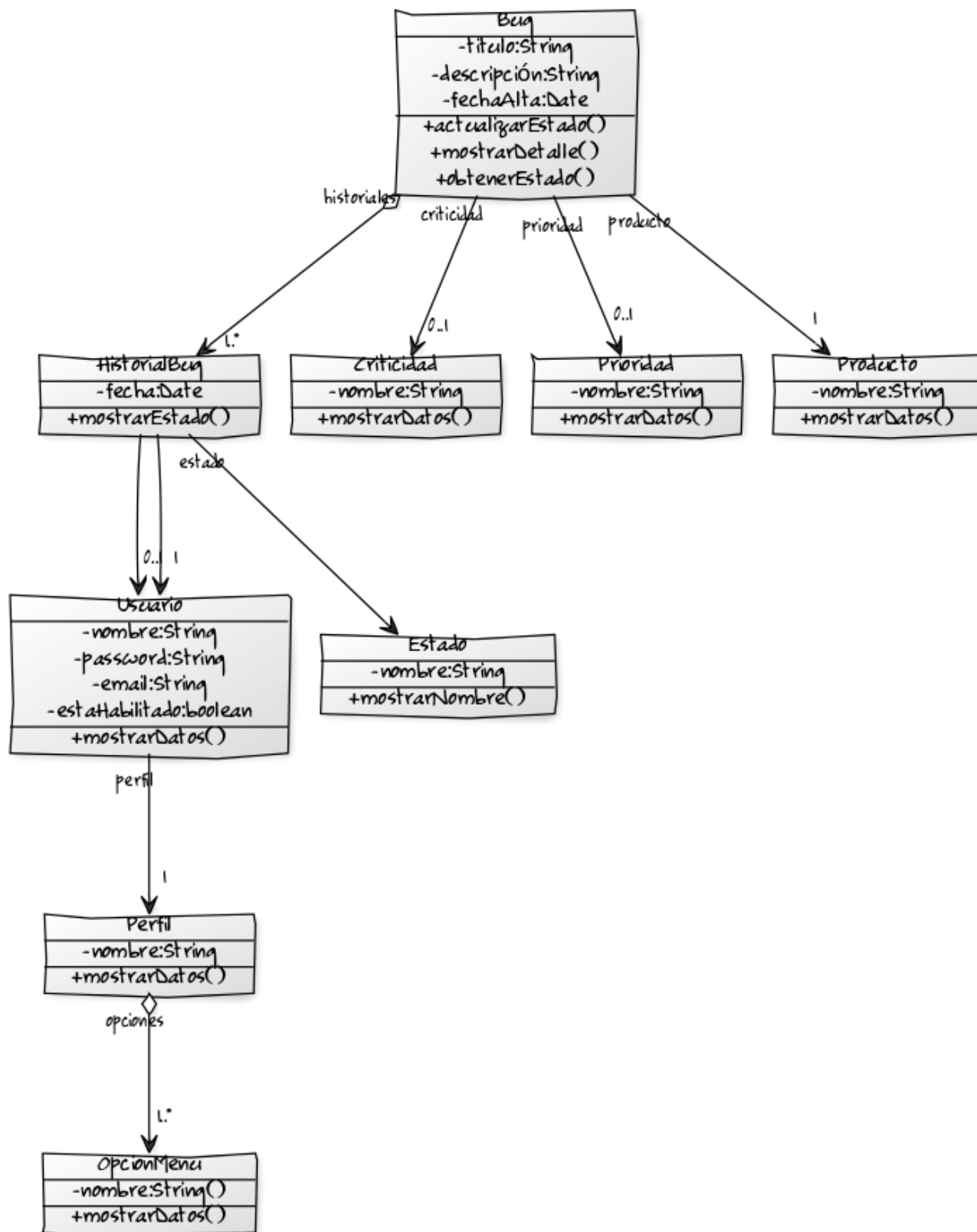
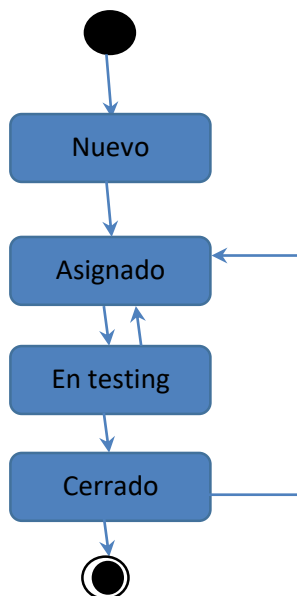
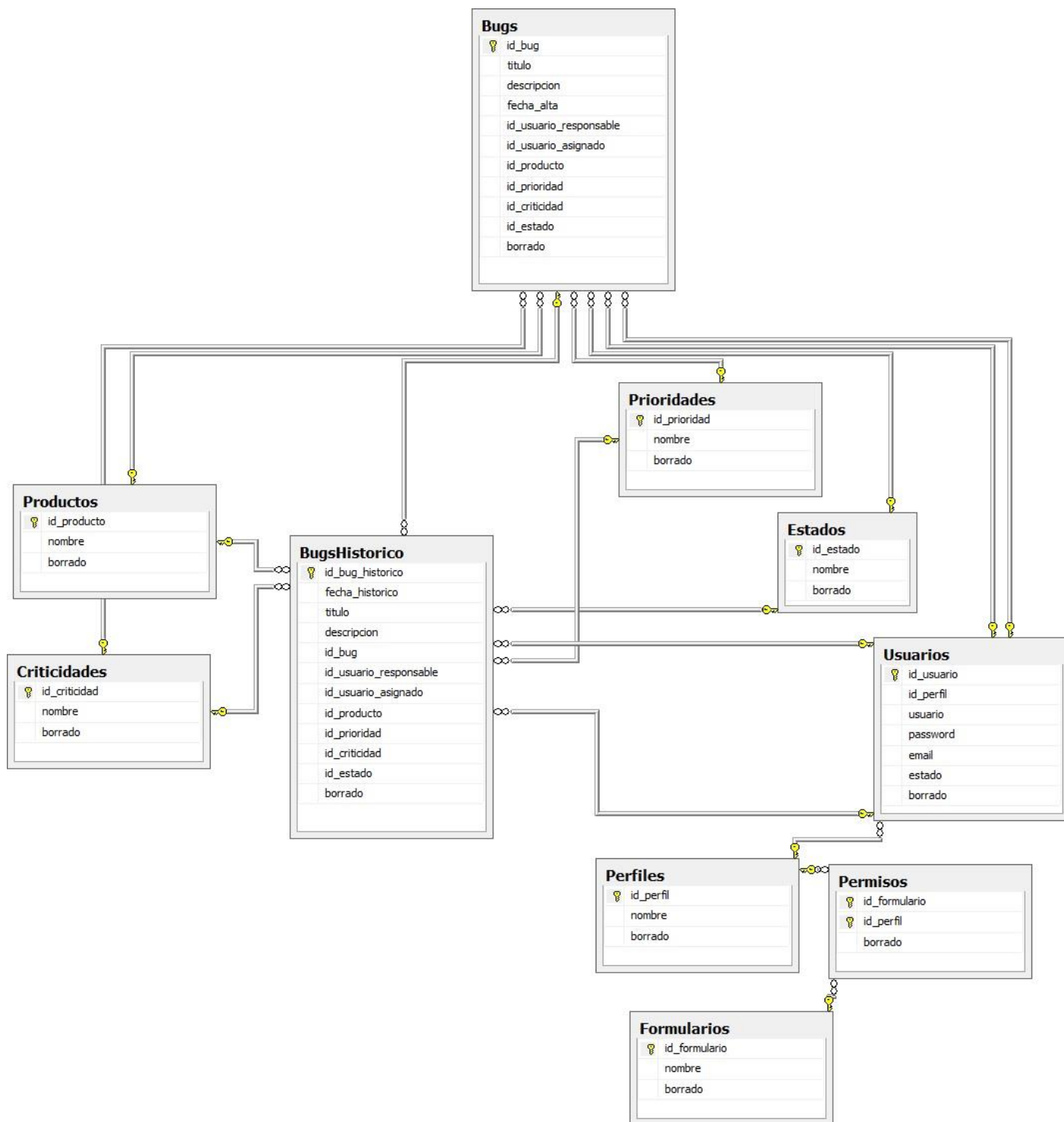


DIAGRAMA DE ESTADOS: Estado del Bug



MODELO DE DATOS ACTUALIZADO





SCRIPT TPI

<https://drive.google.com/file/d/1qNt37YHo2lGTyuJ9ExyfhrnupkSTwu8G/view?usp=sharing>

TRABAJO PRÁCTICO DE SEGUIMIENTO EXTENDIDO

Introducción:

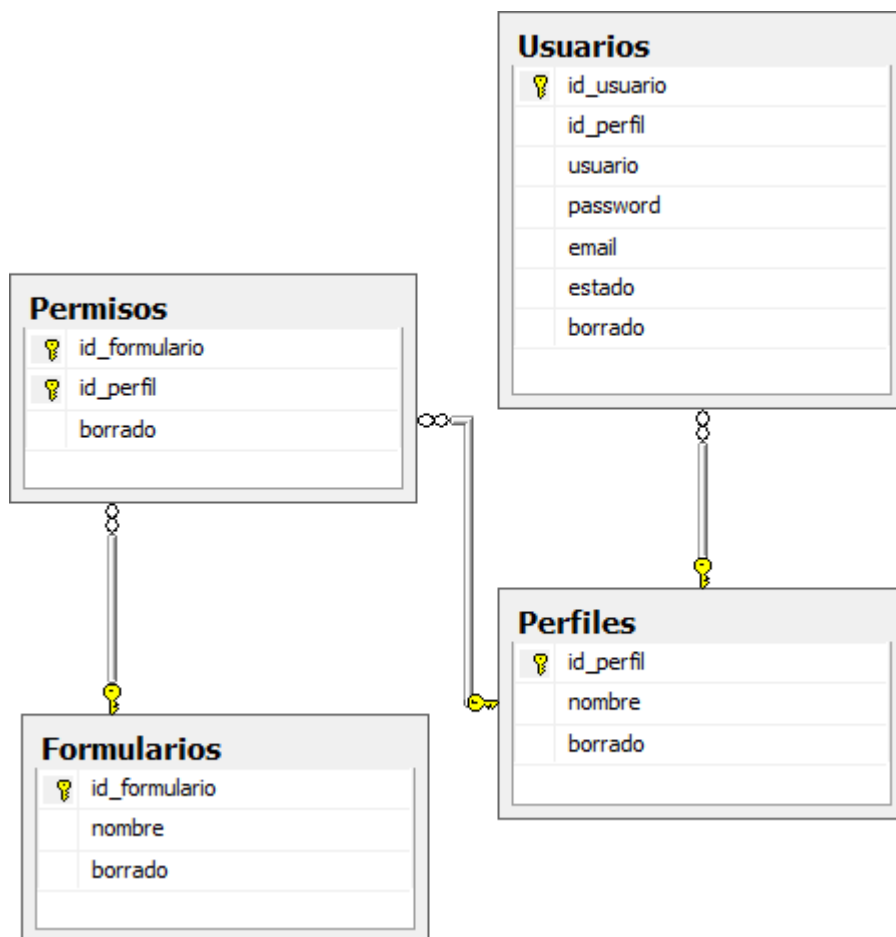
A continuación se presentan los subsistemas propuestos para ser desarrollados en el trabajo grupal. Cada grupo puede elegir alguno de los subsistemas propuestos.

Consideraciones generales:

- Todas las tablas deben soportar borrado lógico (*borrado=true*).
- Todos los trabajos grupales deben contemplar transacciones (*conjunto de actualizaciones en la base de datos que se deben ejecutar todas juntas o ninguna*).
- Se pueden agregar tablas.
- Se debe contemplar la recuperación de borrados.

<i>Script de TPI extendido</i>
https://drive.google.com/file/d/1C5xH7NyUAzZrYC1nXQV6AQQXOWWp4SQI/view?usp=sharing

1) Gestión de usuarios y perfiles (para grupos de 1 o 2 integrantes)

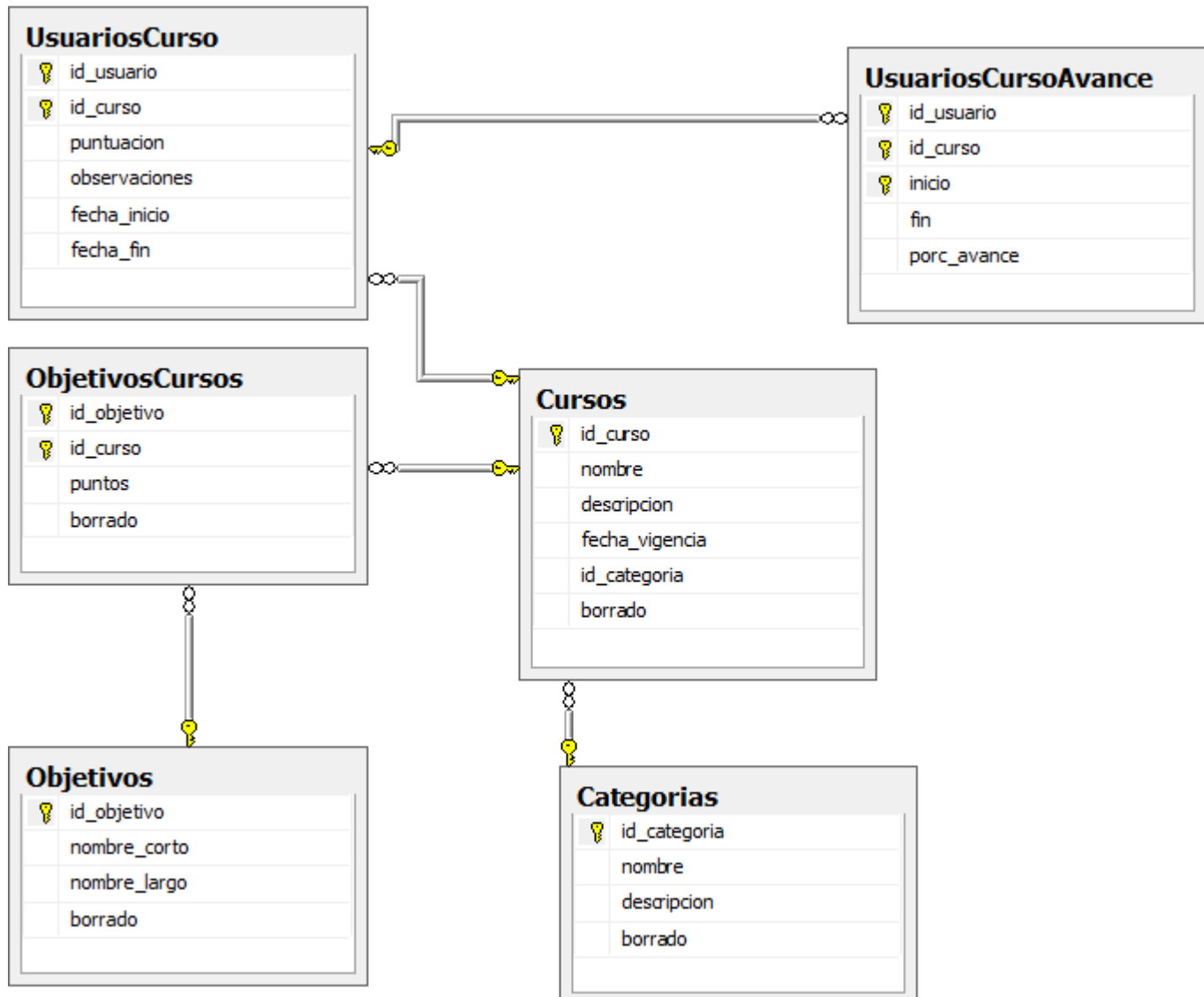


ABMC de perfiles , permisos y formularios.

Transacciones:

- Dado un usuario actualizar los perfiles / permisos asociados a un formulario.
- Dado un usuario mantener el histórico de asignaciones de los perfiles / permisos asociados a un formulario (Agregar al modelo campos y/o tablas necesarios)

2) Gestión de cursos por usuario

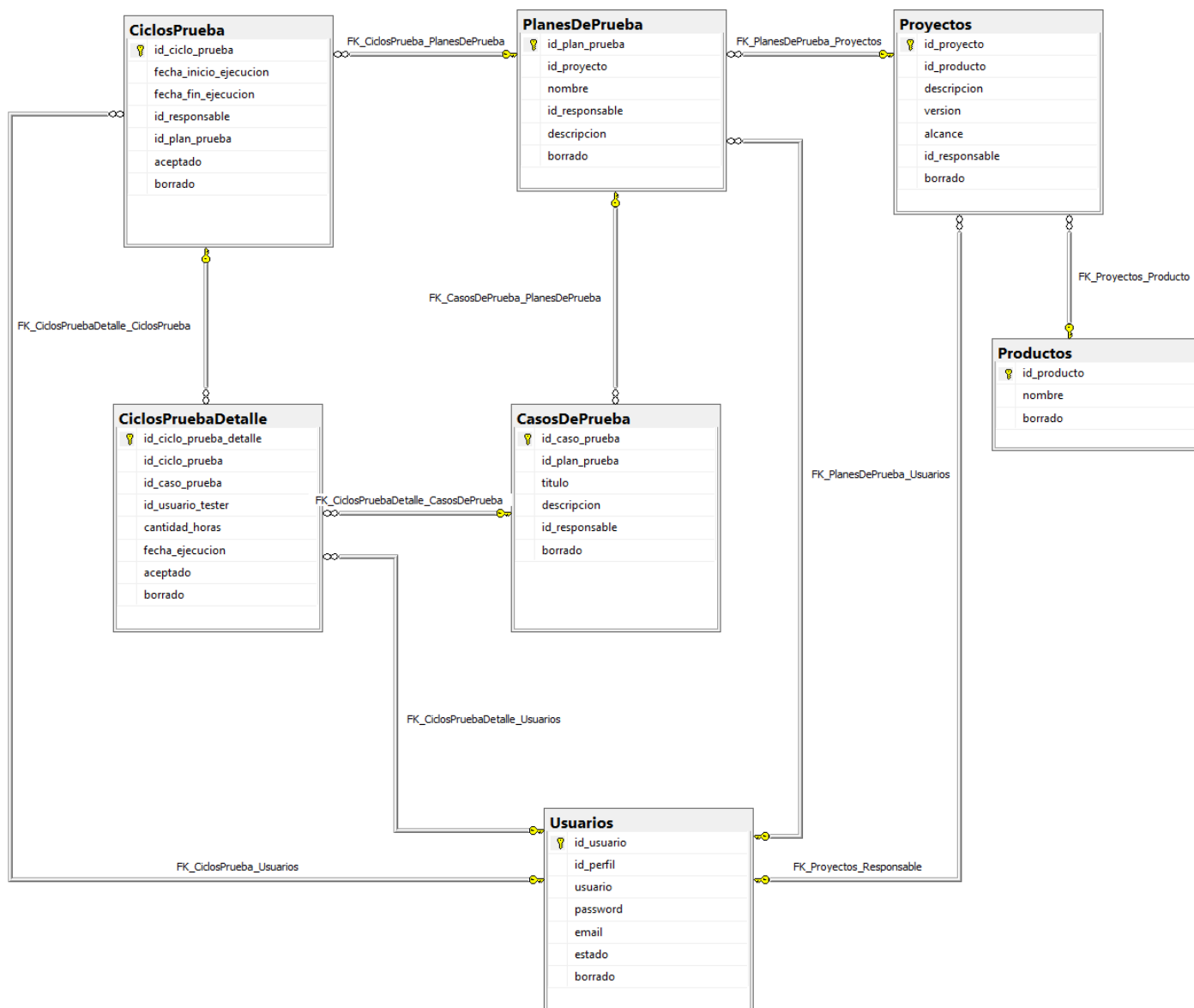


ABMC de cursos, categorías y objetivos

Transacciones propuestas:

- Actualización de todos los objetivos de un curso.
- Actualización del avance de los usuarios por curso.

3) Gestión de productos y planes de testing

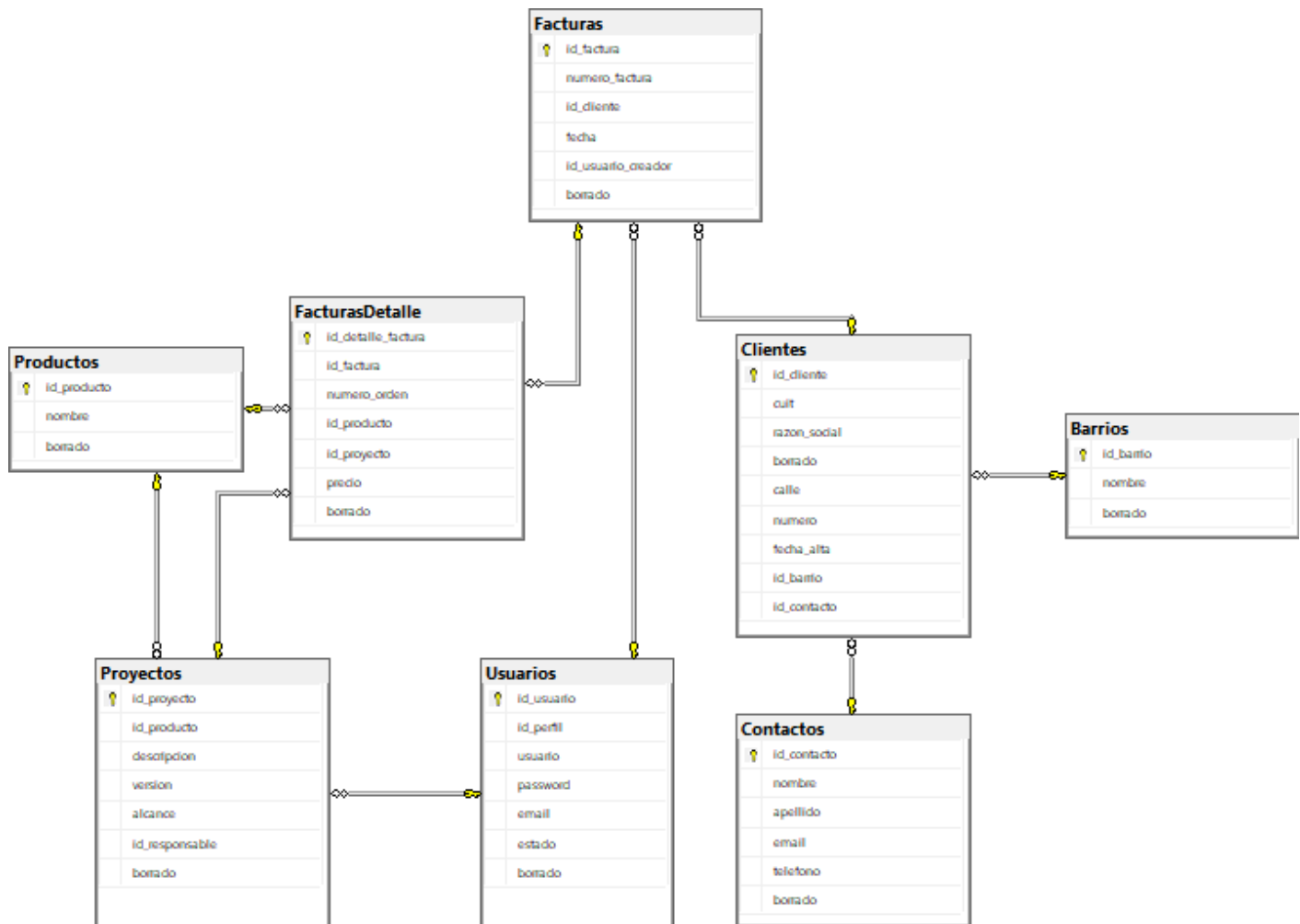


ABM de productos/versiones, casos de prueba, planes de prueba, proyectos.

Transacciones propuestas:

-Composición del detalle de ciclos de prueba para un plan de prueba.

4) Facturación de proyectos y productos

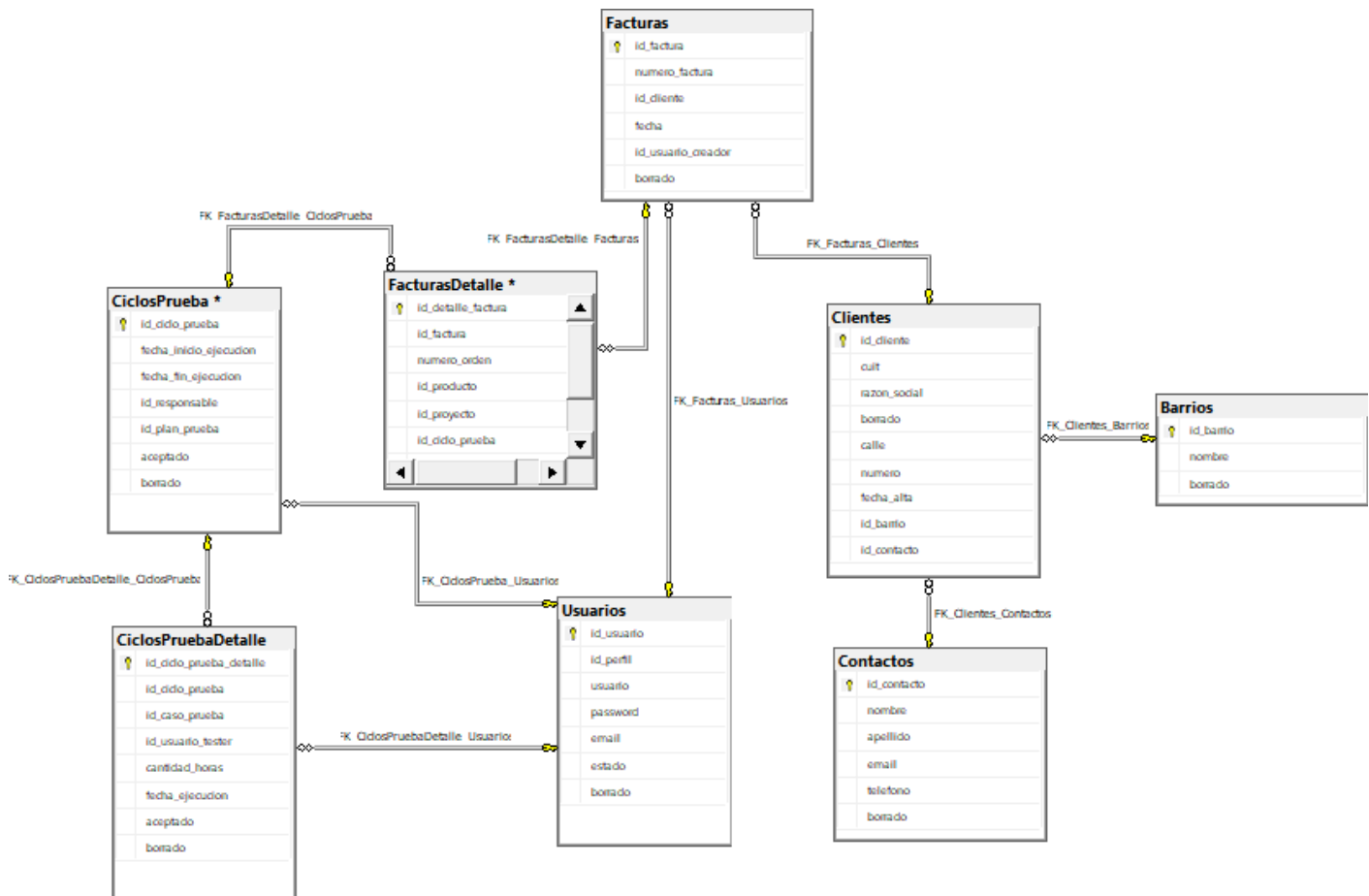


ABM de clientes, barrios, contactos, proyectos

Transacciones:

-Registración de factura: Se puede facturar a un cliente proyectos y / o productos.

5) Facturación de horas de usuarios por plan de prueba

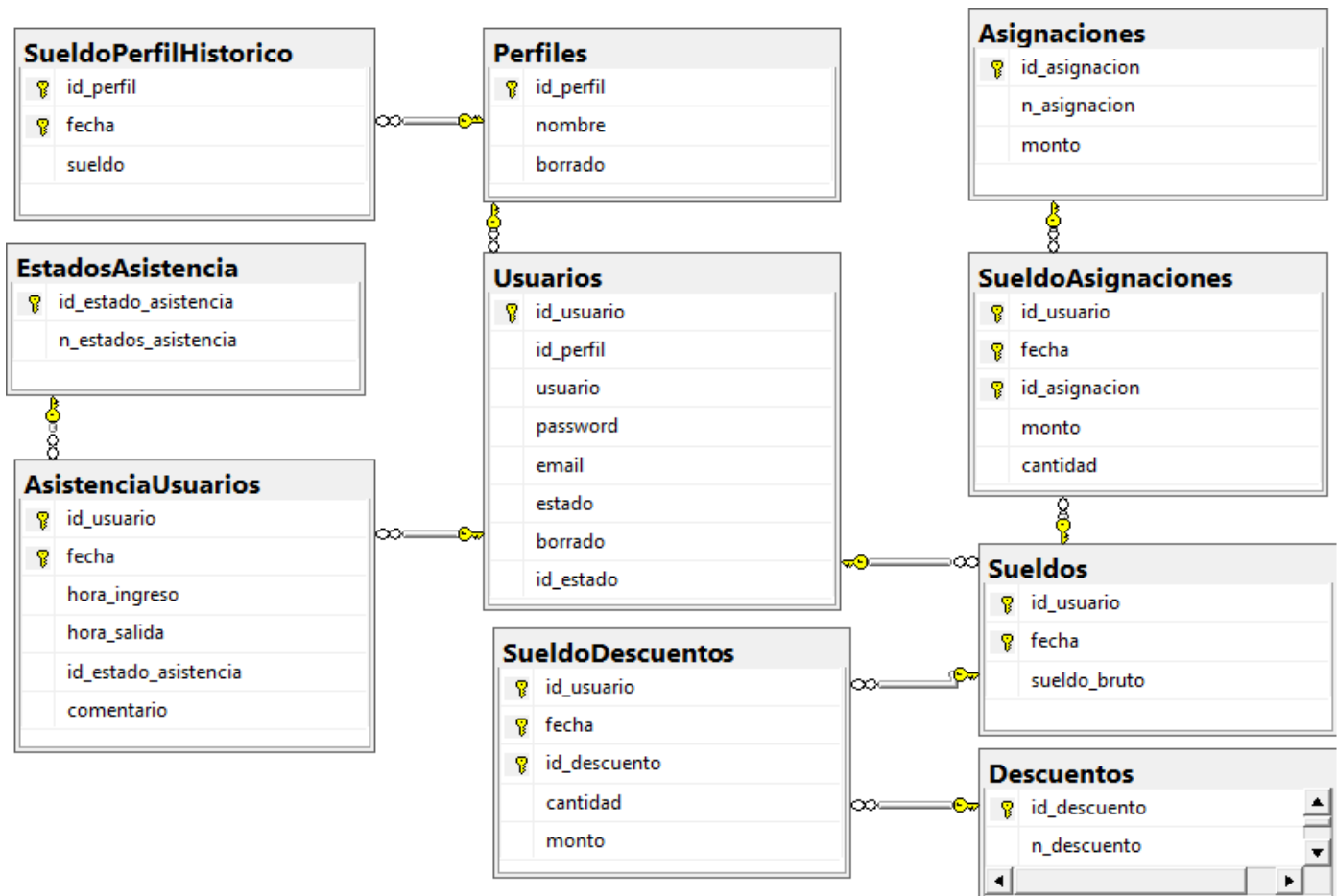


ABM de clientes, barrios, contactos, planes

Transacciones:

-Registración de factura: Se puede facturar a un cliente proyectos y / o productos o las horas invertidas por los testers en distintos ciclos de prueba.

6) Liquidación de sueldos:



ABM de Asignaciones, Asistencia, Descuentos, etc

Transacciones:

Generación mensual de sueldo y su detalle.