**UNIVERSIDAD DON BOSCO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE COMPUTACIÓN**

Diagrama

Descripción generada automáticamente con confianza baja

**Programación con estructura de datos**

**Ing. Santiago José Campos Robles**

**Grupo 04T**

**Tema:**

Fase 3 - Proyecto de catedra

|  |  |
| --- | --- |
| **Alumno** | **Código** |
| Anthony Eduardo Ortega Cruz | OC200384 |
| Hancel Brandon Alvarado Flores | AF202391 |
| Byron Antonio Méndez Hernández | MH192214 |
| Christian Giovanni Tobar Cerón | TC192221 |

10 de mayo de 2021

Contenido

[Contenido 2](#_Toc71462423)

[Introducción 3](#_Toc71462424)

[Objetivos 4](#_Toc71462425)

[Objetivo General 4](#_Toc71462426)

[Objetivos específicos 4](#_Toc71462427)

[Aplicativo de la Estructura de datos seleccionada 5](#_Toc71462428)

[Ambiente de desarrollo 9](#_Toc71462429)

[Login. 10](#_Toc71462430)

[Pruebas de identificación con Roles. 10](#_Toc71462431)

[Mostrar módulos 13](#_Toc71462432)

[Desarrollo de mantenimiento y menús 14](#_Toc71462433)

[Reportes 20](#_Toc71462434)

[Reportes en visual 23](#_Toc71462435)

[Diagrama Entidad-Relación 24](#_Toc71462436)

[Diccionario de datos 24](#_Toc71462437)

[Registros dentro de entidades 28](#_Toc71462438)

[Tabla Usuarios 28](#_Toc71462439)

[Tabla Solicitudes 28](#_Toc71462440)

[Tabla Roles 28](#_Toc71462441)

[Tabla Estados 29](#_Toc71462442)

[Tabla Departamento 29](#_Toc71462443)

[Tabla Casos 29](#_Toc71462444)

[Tabla Bitácora 30](#_Toc71462445)

[Cronograma de actividades 30](#_Toc71462446)

[Conclusión 31](#_Toc71462447)

[Bibliografía 32](#_Toc71462448)

# Introducción

El motivo por el gran interés dedicado a la gestión y creación de proyectos se hace verídico mediante la evaluación actual de los negocios. Reconociendo que la gestión de proyectos es un sinónimo a una gestión de cambio. Hoy en día las empresas u organizaciones buscan modificar su propio enfoque o dirección de tal modo que implementar cambios veraces requerirá de la incorporación de nuevos procesos, productos y de manera infalible en los costos.

Sin embargo, es preocupante ver un gran número de proyectos que no llegan a satisfacer sus objetivos básicos. Por lo que esto no debe de oscurecer la finalidad relacionada a la creación y gestión de proyectos.

La creación y gestión de proyectos tienen como finalidad fundamental la planificación, el control de actividades, materiales y recursos humanos que participan en un desarrollo de un Sistema de Información. Como resultado es posible identificar los problemas que se producen proporcionando una solución de manera inmediata.

Todo proyecto nace de una necesidad, orientado a la obtención de un resultado dentro de un plazo de tiempo reducido con el fin de poder determinar los recursos y su alcance formado por un conjunto especifico de operaciones para alcanzar una meta en común, por ello es habitual que dentro de un equipo de proyecto a menudo se incorporen individuos que no habitualmente no trabajan juntos ya sea por proceder de diferentes áreas.

Esta y otras características hacen que un proyecto deba ser guiado y/o gestionado por especialistas para realizar una entrega a tiempo dentro del presupuesto, un plazo y la calidad con la que se ha planeado además de ofrecer un aporte al aprendizaje. Es donde nace la importancia de una buena gestión de proyectos que recurre a la aplicación de conocimientos, técnicas y habilidades para la realización de un proyecto eficiente y eficaz.

# Objetivos

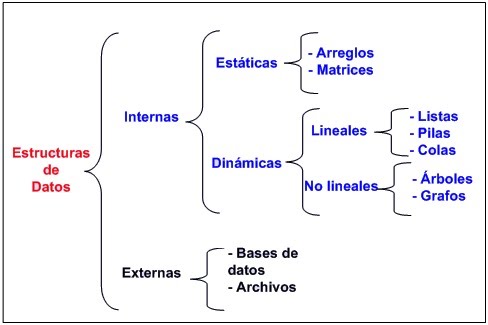
## Objetivo General

Crear una solución que ayude a las empresas u organizaciones en la gestión de proyectos de desarrollo, tanto para el inicio de uno nuevo, como para la realización de mejoras de uno existente, por medio del desarrollo de una herramienta de software.

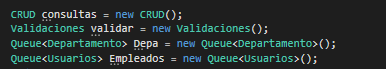
## Objetivos específicos

* Definir una serie de pasos para la gestión de proyectos de desarrollo de software con el fin de prevenir una mala planificación y organización a lo largo del seguimiento del requerimiento.
* Mejorar las retroalimentaciones de avance en los proyectos, para evitar inconvenientes por una mala comunicación de las etapas que ya están finalizadas.

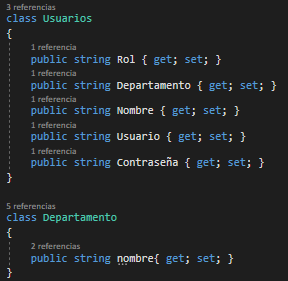
## Aplicativo de la Estructura de datos seleccionada



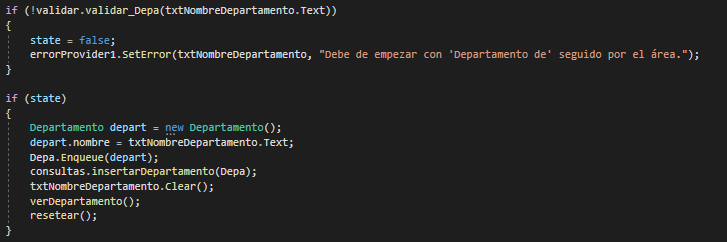
Creación de colas que se utilizaran



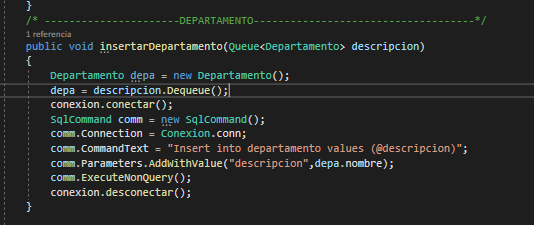
Clases enfocadas a la estructura de datos utilizada

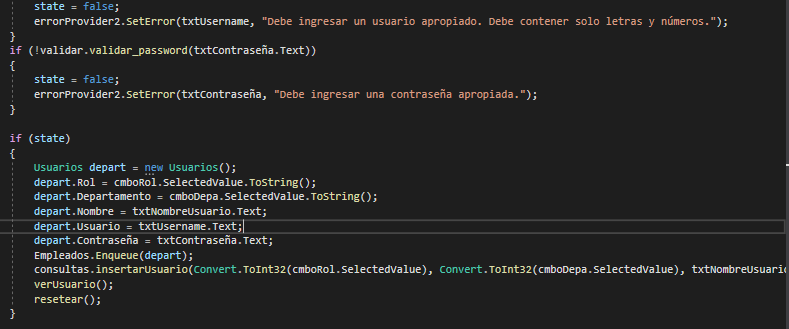


Proceso de Encolar la clase destinada

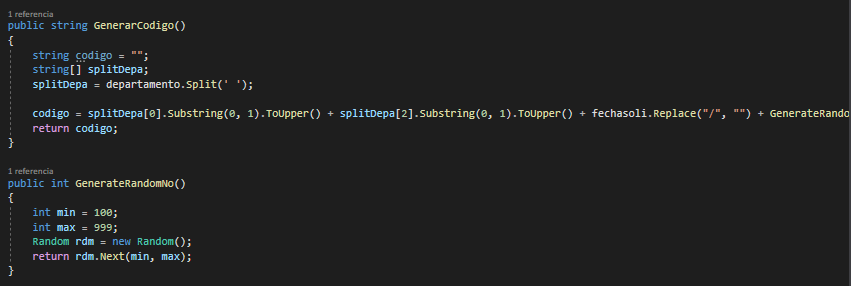


Proceso de desencolamiento e inserción de datos en la BDD SQL Server, los datos pasas como parámetros con sus valores y se agregan junto al commandText para luego ejecutar la consulta y guardarse los datos el gestor de base de datos.

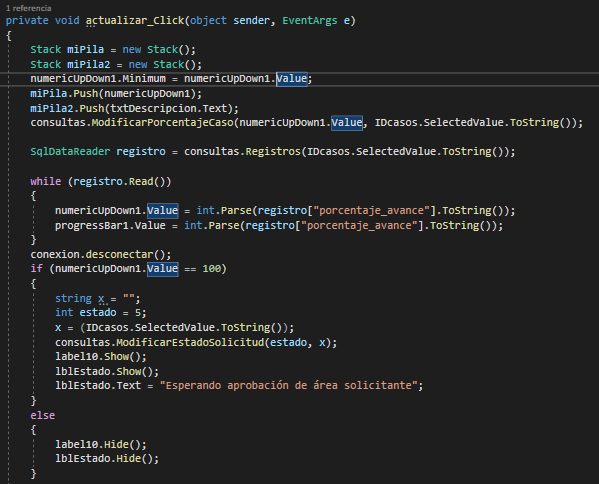




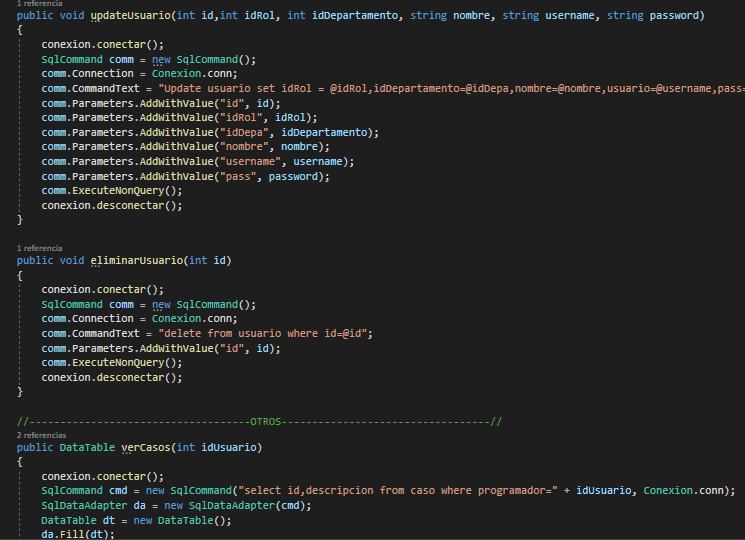
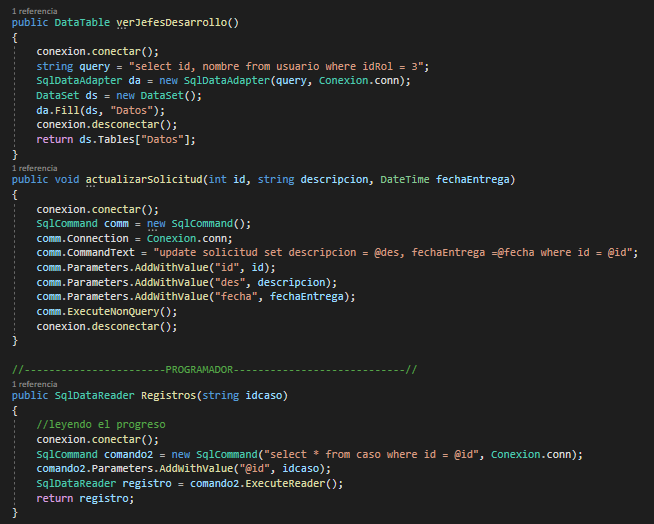
Uso de arreglos como estructuras estáticas para la generación de códigos al momento de aceptar nueva solicitud de caso.



Uso de pilas para guardar datos sobre bitácoras y porcentajes de proyectos.



Consultas SQL



## Ambiente de desarrollo

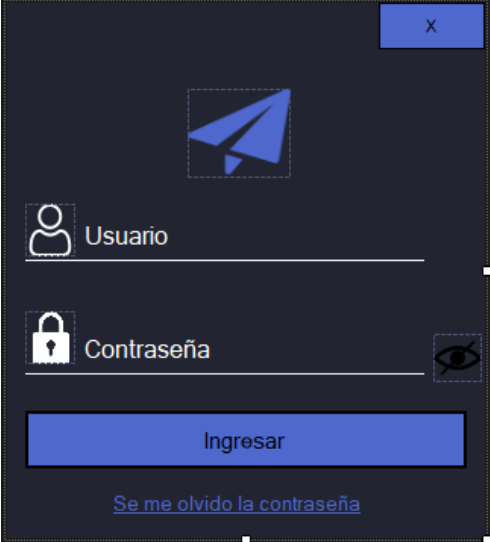
Como grupo hemos visto a bien utilizar el ambiente de formularios parar realizar a cabo el proyecto, principalmente por las facilidades de personalización y la amplia variedad de herramientas que ofrece esta interfaz para el desarrollo de tareas específicas que se desarrollan dentro del proyecto.

A esto le sumamos que al trabajar de esta manera nos percatamos como se desenvuelve el proyecto de cara a los usuarios de este, ya que el ambiente de formularios da un ambiente familiar y amigable a los usuarios de aplicaciones de escritorio.

Nosotros como grupo hemos llegado a decidir que el ambiente más óptimo para el desarrollo de nuestro proyecto es en **Ambiente de escritorio “más concretamente en una Ambiente de Formularios”** ya que nos brinda las herramientas y oportunidades de dar a lucir nuestro proyecto de **CASOLITYCAL** simulando la implementación de dicho proyecto ya en un entorno laboral sin dejar de lado que previamente analizamos los pros y contras que trae el desarrollarlo en un ambiente como el que hemos escogido y a su vez sumarle el objetivo que tenemos y es el implementar las estructuras de datos dándoles un buen e importante uso dentro de la aplicación lo cual en un futuro nos facilite el mantenimiento y control de datos dentro del sistema, también todos los miembros participantes llegamos a ciertas conclusiones al optar el desarrollo en un ambiente de escritorio y ocupar un ambiente de formularios ya que una app desarrollada aquí brinda mayor robustez y un mejor tiempo de respuesta así también abre un mar de posibilidades futuras al pensar que nos da la facultad de realizar cualquier cambio a cosas siempre y cuando el software nos lo permita (cuestión gráfica, control total de las entradas del usuario al momento de capturar) de igual modo no podemos dejar de lado que entre sus desventajas más notorias esta que la portabilidad de la aplicación no es la mejor ya que no es conveniente si nuestra empresa contara con muchos usuarios los cuales estarán accediendo desde diferentes dispositivos pero a pesar de todo eso planeamos como empresa y futuros programadores desarrollar –dar un buen uso e implementación de las estructuras de datos dándole a si al usuario final la mejor experiencia al visitar nuestro proyecto

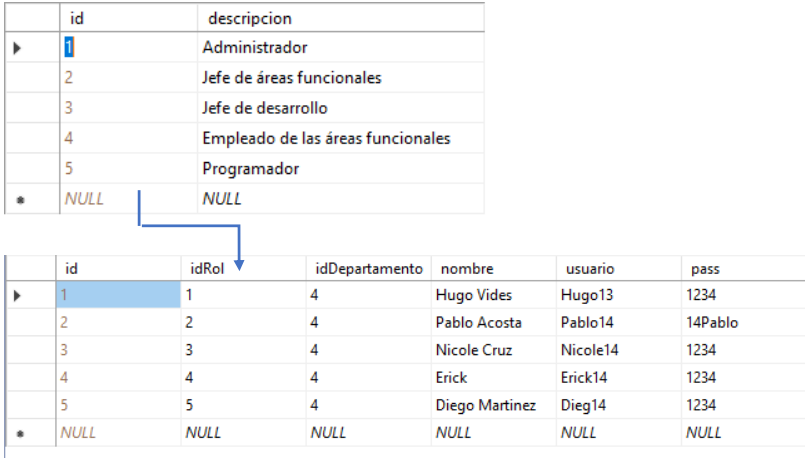
## Login.

El inicio de sesión se realizará con los datos registrados por el usuario administrador en el se ingresará en nombre de usuario y la contraseña, si el usuario desea ver la contraseña podrá presionar sobre el botón del lado derecho.

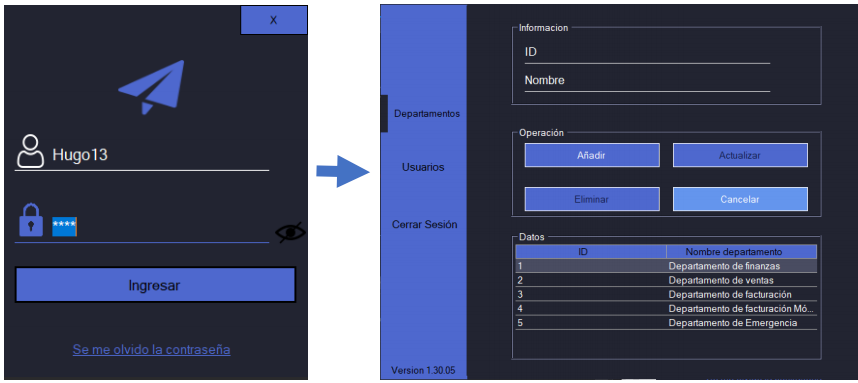


## Pruebas de identificación con Roles.

Con la siguiente tabla de usuario se accederá y así se dará privilegios de acuerdo a su idRol.

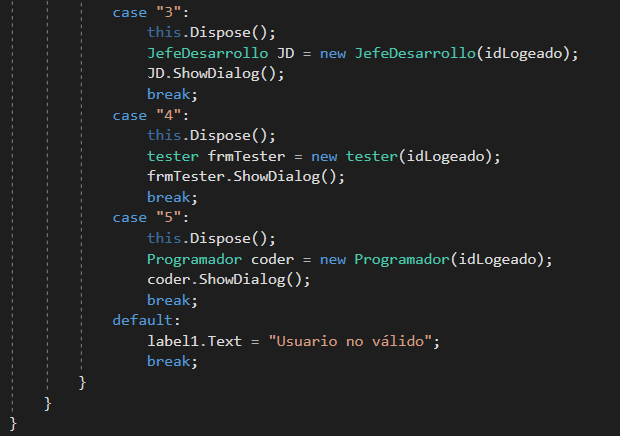
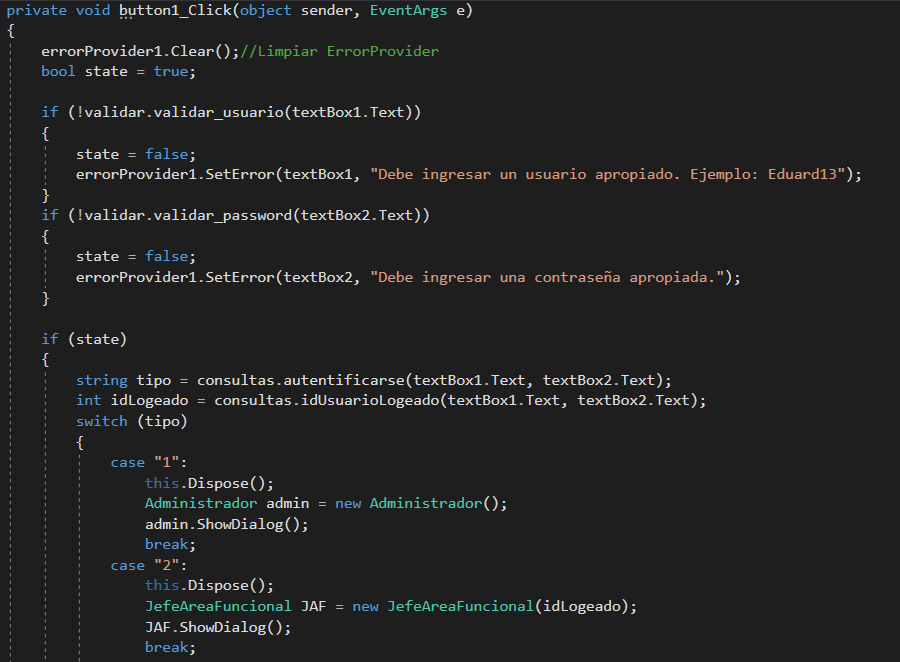


Accediendo como Administrador



Accediendo como jefe de áreas funcionales





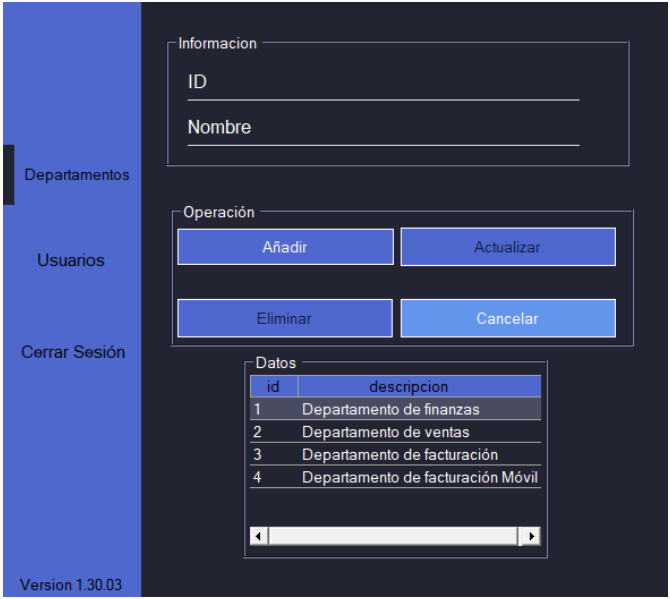
# Mostrar módulos

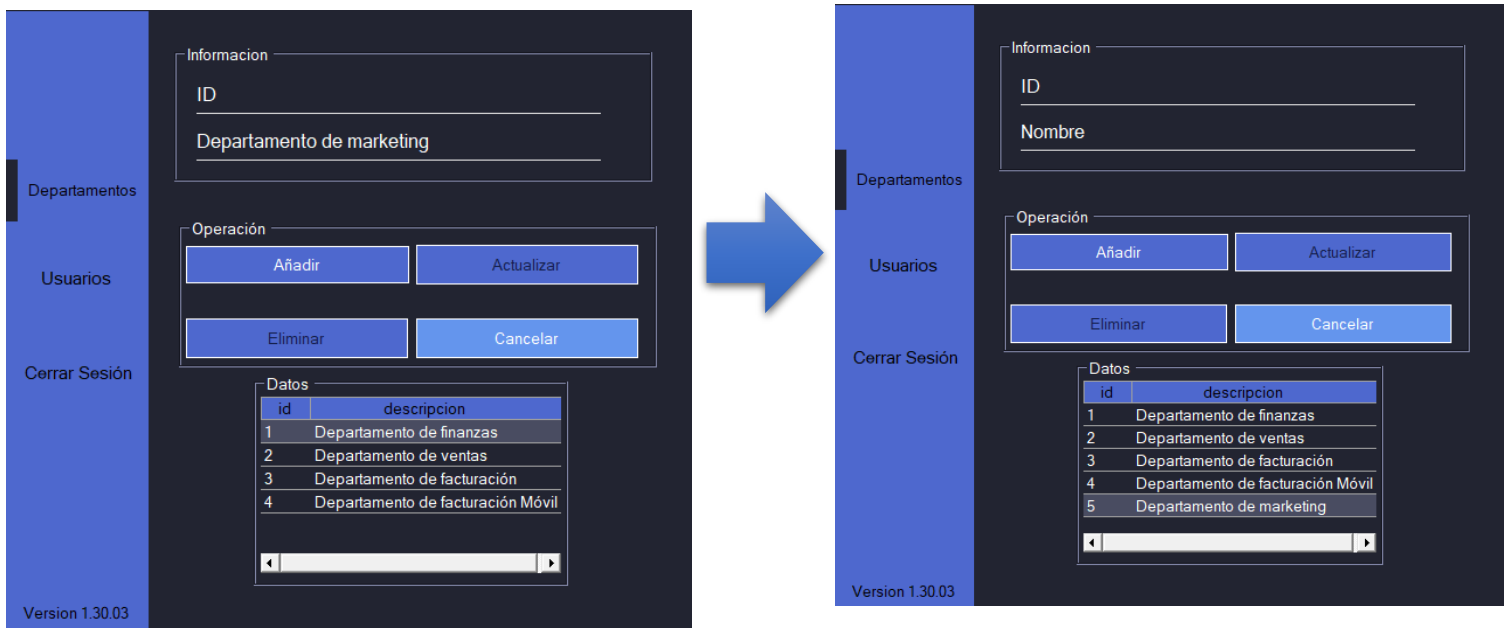
* Modulo administrador  
  Con la capacidad de registrar y gestionar áreas funcionales de la empresa (departamentos), jefes de áreas funcionales y jefes de desarrollo.
* Modulo jefes de áreas funcionales  
  Con la capacidad de solicitar la apertura de casos y monitorear el porcentaje de progreso y bitácora de los casos aperturados.
* Modulo empleados de las áreas funcionales  
  Pueden ser asignados como “probadores” de un caso, en tal situación deben monitorear el porcentaje de progreso y bitácoras de trabajo de dicho caso y aprobarlo o rechazarlo una vez que este haya sido entregado por el programador asignado.
* Modulo jefe de desarrollo   
  Tiene la capacidad de aceptar o rechazar las solicitudes de casos realizadas por los jefes de las áreas funcionales que tiene a su cargo. Además, debe tener la capacidad de monitorear el trabajo de los programadores que tiene a su cargo.
* Modulo programadores  
  Debe actualizar los porcentajes de progreso y bitácoras de trabajo de los casos a los que ha sido asignado.

# Desarrollo de mantenimiento y menús

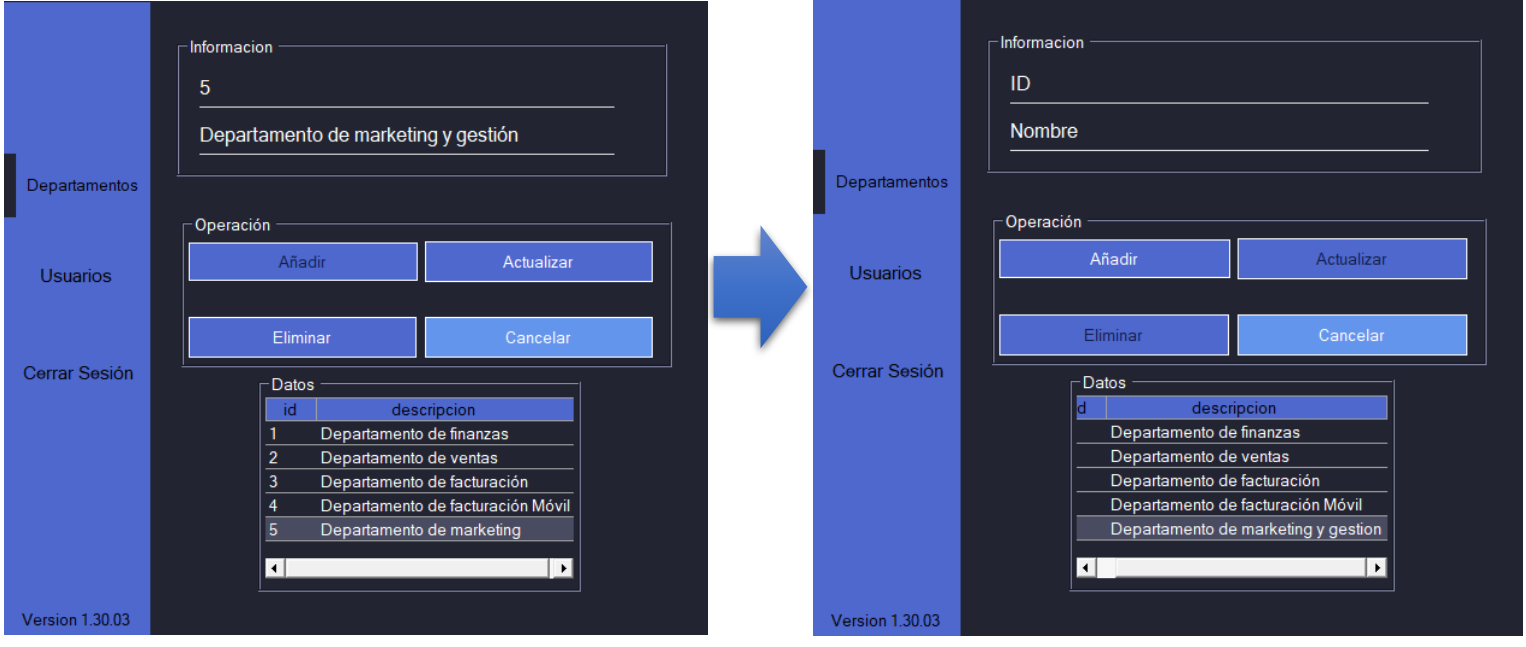
Departamentos

El usuario tipo “Administrador” tendrá acceso a la creación de departamentos, ejemplo “Departamento de marketing” dentro de cada departamento tendrá su tipo de área de trabajo y administración de proyectos. El administrador registra el nombre del departamento y el sistema genera un código o ID auto incrementable.

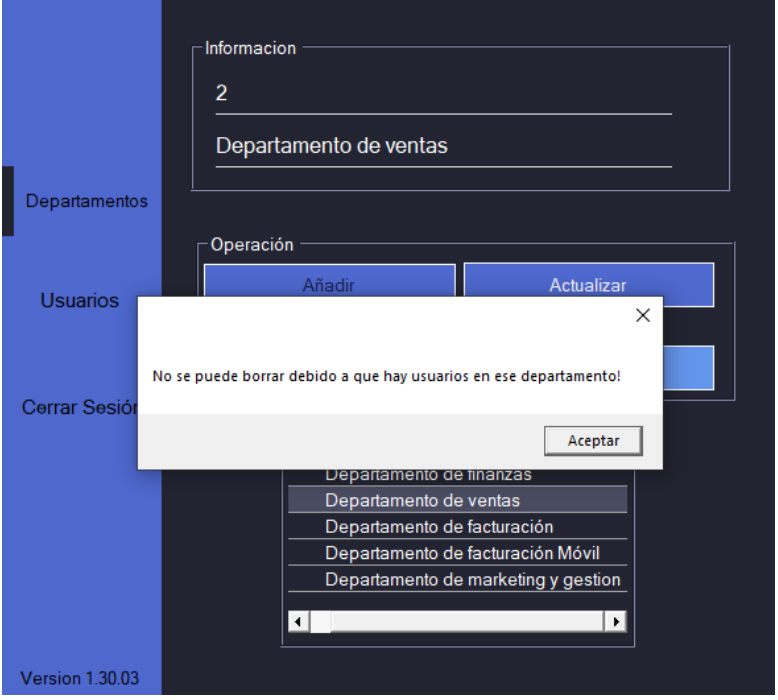




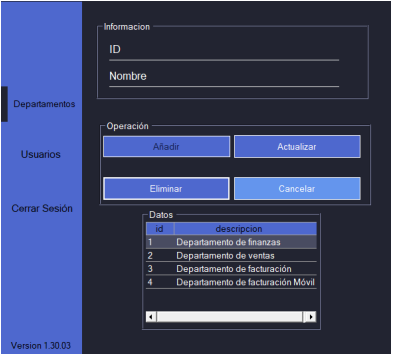
Para editar los departamentos que se encuentran dentro del datagridview el usuario “Administrador” deberá dar doble click en la fila que desea modificar. Luego realizar los cambios según el nombre y presionar el botón “Actualizar”.



Finalmente, si el usuario “Administrador” decide eliminar cualquier departamento este podrá siempre y cuando no existan otro tipo de usuario dentro de él, caso contrario mostrará un mensaje de error.



Para eliminar un departamento se requiere de la misma función realizada al momento de editar un departamento, en este caso deberá dar doble click en la fila o departamento a eliminar y presionar el botón “Eliminar” para ejercer los cambios.

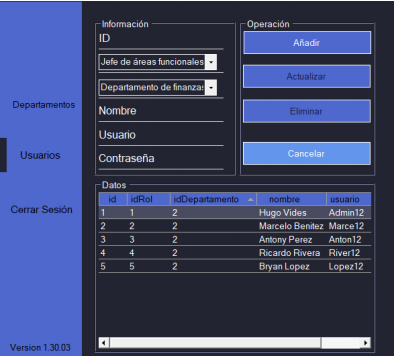


Para cancelar cambios al momento que el usuario “Administrador” haya seleccionado solo deberá presionar en el botón “Cancelar” para limpiar datos escritos en los textbox

Usuarios

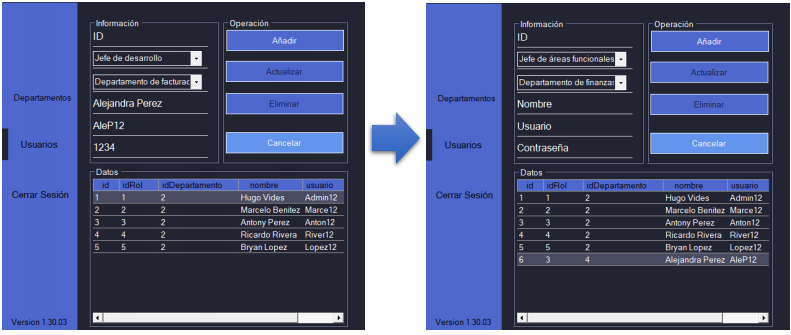
El usuario “Administrador” podrá añadir más usuarios con los siguientes datos:

* Tipo de usuario.
* Departamento (explicado en el punto anterior).
* Nombre de la persona.
* Nombre de usuario o Username.
* Contraseña.



El usuario “Administrador” podrá agregar más usuarios administradores, pero no podrá modificarlos, mostrará un mensaje de advertencia. Para añadir más usuarios el usuario encargado ingresará los datos mencionados con anterioridad y presionará en el botón añadir, los botones de actualizar y eliminar se encuentran bloqueados por el hecho que no se encuentran usuarios seleccionados. Cada campo contará con ciertas validaciones que deberán cumplirse.

* El nombre solo debe incluir caracteres de letras, sin incluir números ni símbolos.
* El usuario deberá incluir letras y números un mínimo de 6 caracteres y un máximo de 15 caracteres a escribir.
* La contraseña debe incluir de 4 a 8 caracteres alfanuméricos.



Para editar los usuarios que se encuentran dentro del datagridview, el usuario “Administrador” deberá dar doble click en la fila que desea modificar. Luego realizar los cambios según el nombre de persona, nombre de usuario o username y contraseña para después presionar el botón “Actualizar” de igual forma contará con las validaciones mencionadas con anterioridad.



Para eliminar un usuario se requiere de la misma función realizada al momento de editar un usuario, en este caso deberá dar doble click en la fila o usuario a eliminar y presionar el botón “Eliminar” para ejercer los cambios.



Para cancelar cambios al momento que el usuario “Administrador” haya seleccionado solo deberá presionar en el botón “Cancelar” para limpiar datos escritos en los textbox. Para concluir con el apartado de administrador este tipo de usuario podrá Cerrar sesión a lo cual lo enviará al formulario de Inicio de sesión.

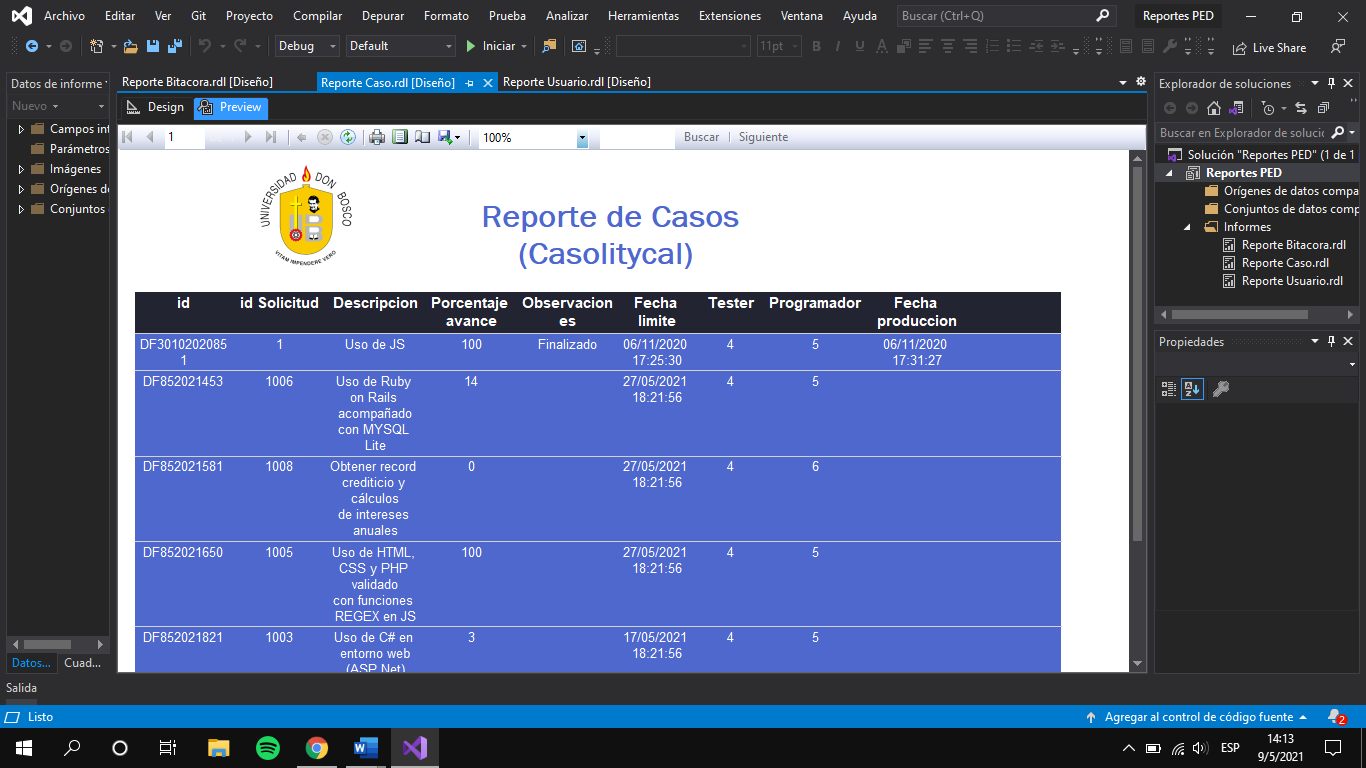
# Reportes

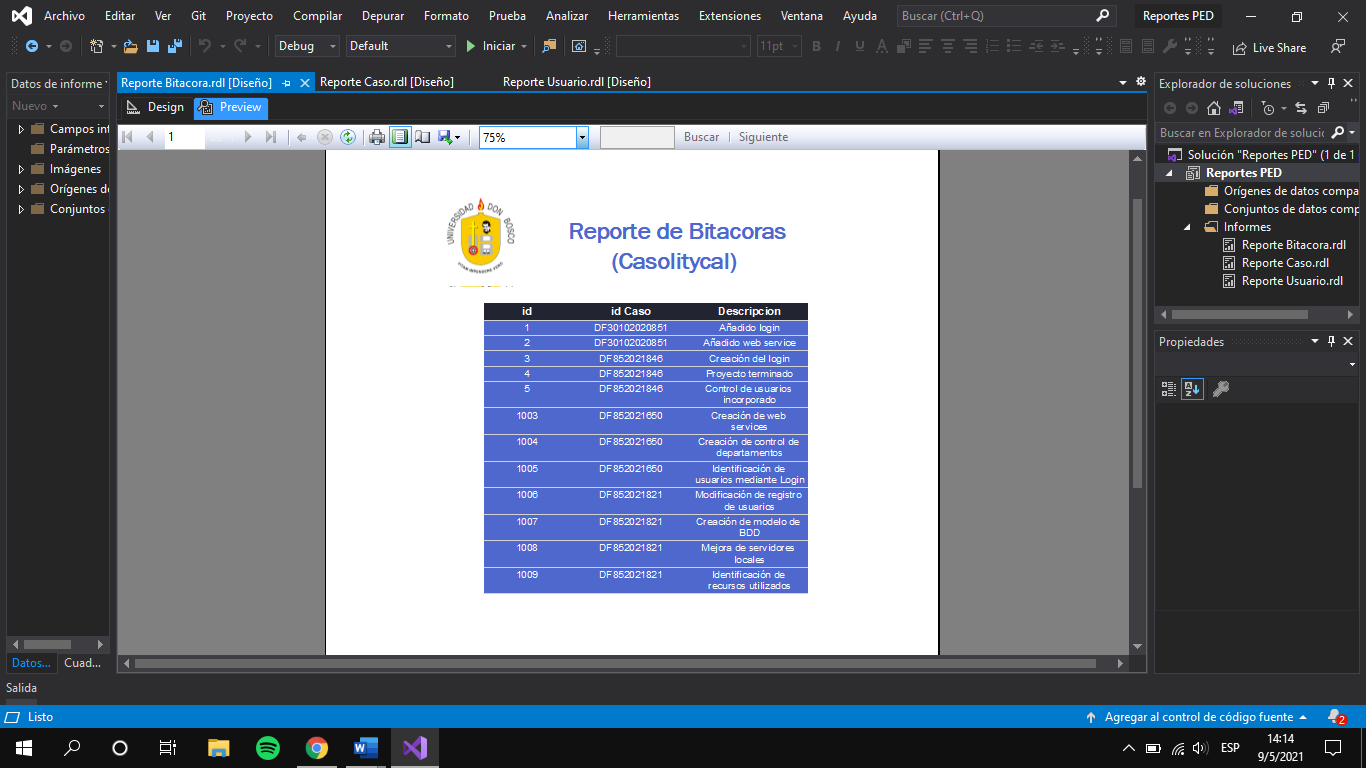
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  | |  |  |  |  |
|  |  | |  | | --- | | **Reporte de Bitácoras**  **(Casolitycal)** | |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | |  |  |  | | --- | --- | --- | | **id** | **id Caso** | **Descripcion** | | 1 | DF30102020851 | Añadido login | | 2 | DF30102020851 | Añadido web service | | 3 | DF852021846 | Creación del login | | 4 | DF852021846 | Proyecto terminado | | 5 | DF852021846 | Control de usuarios incorporado | | 1003 | DF852021650 | Creación de web services | | 1004 | DF852021650 | Creación de control de departamentos | | 1005 | DF852021650 | Identificación de usuarios mediante Login | | 1006 | DF852021821 | Modificación de registro de usuarios | | 1007 | DF852021821 | Creación de modelo de BDD | | 1008 | DF852021821 | Mejora de servidores locales | | 1009 | DF852021821 | Identificación de recursos utilizados | | | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | |  | | --- | | **Reporte de Usuarios**  **(Casolitycal)** | |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **id** | **id Rol** | **Id Departamento** | **Nombre** | **Usuario** | **Contraseña** | | 1 | 1 | 4 | Hugo Vides | Hugo13 | 1234 | | 2 | 2 | 4 | Pablo Acosta | Pablo14 | 14Pablo | | 3 | 3 | 4 | Nicole Cruz | Nicole14 | 1234 | | 4 | 4 | 4 | Erick | Erick14 | 1234 | | 5 | 5 | 4 | Diego Martinez | Dieg14 | 1234 | | 6 | 5 | 5 | David Perez | David02 | 1234 | | | | |  |

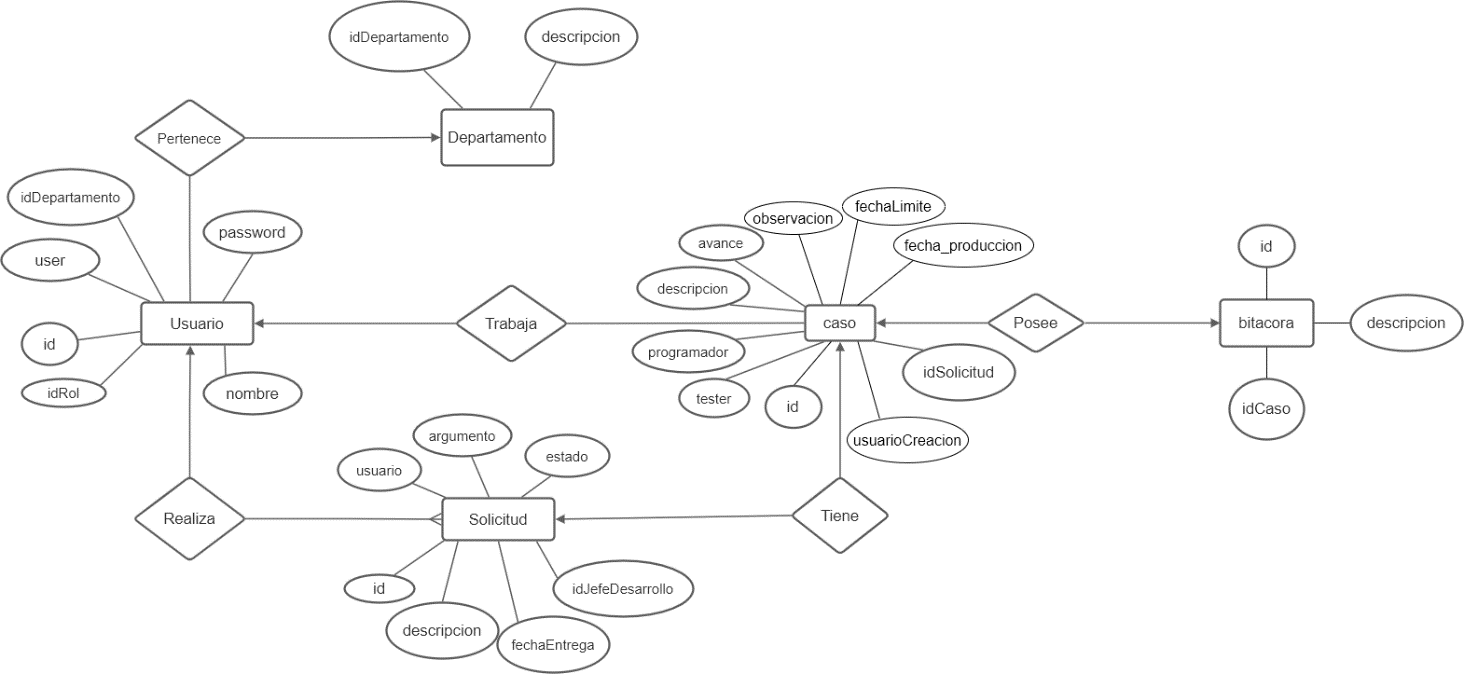
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  | |  | | --- | | **Reporte de Casos**  **(Casolitycal)** | |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **id** | **id Solicitud** | **descripción** | **Porcentaje avance** | **Observaciones** | | **Fecha limite** | **Tester** | **Programador** | **Fecha produccion** |  | | DF30102020851 | 1 | Uso de JS | 100 | Finalizado | 06/11/2020 17:25:30 | | 4 | 5 | 06/11/2020 17:31:27 |  | | DF852021453 | 1006 | Uso de Ruby on Rails acompañado con MYSQL Lite | 14 |  | 27/05/2021 18:21:56 | | 4 | 5 |  |  | | DF852021581 | 1008 | Obtener record crediticio y cálculos de intereses anuales | 0 |  | 27/05/2021 18:21:56 | | 4 | 6 |  |  | | DF852021650 | 1005 | Uso de HTML, CSS y PHP validado con funciones REGEX en JS | 100 |  | 27/05/2021 18:21:56 | | 4 | 5 |  |  | | DF852021821 | 1003 | Uso de C# en entorno web(ASP.Net) | 3 |  | 17/05/2021 18:21:56 | | 4 | 5 |  |  | | DF852021846 | 3 | -Trabaje en SQL SERVER -JS, PHP y HTML | 100 | Finalizado | 15/05/2021 15:16:28 | | 4 | 6 | 16/06/2021 15:17:29 |  | | DF852021918 | 1004 | Uso de JS | 83 |  | 22/05/2021 18:21:56 | | 4 | 5 |  |  | | | | | |  |

## Reportes en visual





# Diagrama Entidad-Relación



# Diccionario de datos

El diseño de la base de datos estarás representado por 7 tablas las cuales cada una tendrá una función diferente:

* Caso.
* Solicitud.
* Bitácora
* Departamento
* Usuario.
* Rol
* Estados

El nombre de las clases les fue colocado teniendo en cuenta qué es lo que cada una de ellas realizará dentro del proyecto, para detallar un poco más estas empezamos con la clase Usuario.

* **Tabla rol:** en ellas representamos los diferentes roles que tendrá cada uno de los usuarios:

1. Id: identificador único de cada rol que se tenga dentro del proyecto.
2. Descripción: nombre del campo otorgado ejemplo: Administrador, programador, etc

**Tabla Estado:** contendrá los diferentes avisos 0 estados que dispondrá en solicitudes, casos, bitácoras, etc.

1. Id\_estado: identificador que tendrá la tabla estado
2. Nombre: descripción del caso. Ejemplo: Vencido, en espera, etc.

* **Tabla Usuario:** en ella representamos a los diferentes tipos de usuarios que tienen acceso al sistema, se guardan principalmente datos generales de una persona los que definimos en sus atributos:

1. Id: identificador único de cada usuario que se registre dentro del proyecto.
2. idDepartamento: este nos permite conocer de forma fácil y rápida saber a qué departamento pertenece ese usuario.
3. idRol: identificador encargado de referirnos a los tipos de roles dentro del programa y describir qué tipo de usuario es la persona registrada.
4. Nombre: nombre del empleado registrado sin importar su rol.
5. Usuario: es el medio con el cuál se identifican antes de entrar al programa.
6. Password: contraseña registrada que le da acceso a cada usuario cuando quieran hacer uso del sistema.

Esta clase no cuenta con métodos, ya que solo describe las generalidades de una persona, en las especializaciones de esta clase son las que detallan qué funciones tiene dentro de la plataforma.

* **Tabla Departamento:** Clase encargada de representar a los departamentos que están dentro de la empresa, no realizan ninguna función dentro del programa sus atributos son:

1. idDepartamento: Identificador único de cada uno de los departamentos.
2. Descripción: contiene el nombre del departamento.

* **Tabla Solicitud:** clase que almacena y describe las solicitudes hechas.

1. Id: identificador único asociado a cada una de las solicitudes creadas.
2. Descripción: contiene el detalle de lo que se ha solicitado, describe la problemática y las necesidades necesitan ser cubiertas.
3. fechaEntrega: es la fecha en que se crea la solicitud.
4. idJefeDesarrollo: identificador del jefe de desarrollo al que se le asignó la solicitud.
5. Estado: la disponibilidad de la solicitud, describe si esta está en espera, rechazada o aprobada.
6. usuarioCreacion: el id del jefe de área funcional que solicitó el desarrollo de la solución.
7. Argumento: en caso de ser rechazada la solicitud se debe de poner una razón por la cual se ha rechazado la solicitud.

* **Tabla Caso:** Esta clase guarda el detalle del avance de las soluciones que estén en desarrollo.

1. Id: identificador único de cada uno de los casos con un formato específico (3 letras, la fecha y tres números), la fecha en que se realizó la solicitud formato (YY) y un número de tres cifras generado aleatoriamente.
2. idSolicitud: es el identificador de la solicitud de la cuál proviene el desarrollo de la solución.
3. Descripción: descripción de la solicitud detallada con los comentarios del jefe de desarrollo.
4. porcentajeAvance: campo que registra el avance de desarrollo de las soluciones.
5. fechaLimite: Viene dado por el jefe de desarrollo, es la fecha en la que la solución debe de ser entregada al tester designado.
6. usuarioCreacion: almacena el identificador del usuario que crea el caso.
7. Tester: el campo almacena la referencia del usuario designado como tester una vez esté terminada la solución solicitada.
8. Programador: se refiere al id del programador que el jefe de desarrollo puso como encargado del proyecto.
9. Fecha\_produccion: se registra a la fecha en la que se acepta la solución y se pone a disposición del departamento.

* **Tabla bitácora:** Contiene los cambios realizados por los programadores, sobre el avance de los proyectos.

1. Id: identificador de las bitácoras asociado a cada una de ellas.
2. idCaso: se refiere al id del caso al cual pertenece la bitácora.
3. Descripción: es el campo donde se actualiza periódicamente el avance del proyecto.

**Relaciones:**

Entre la clase usuario y sus hijas existe herencias, desde la clase padre Usuario hacia las demás clases derivadas o las especializaciones de esta misma.

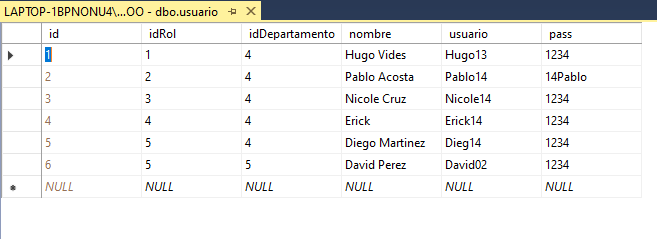
La relación que existe entre Departamento y Usuario es de muchos a uno, donde muchos usuarios pertenecen a un departamento y un departamento puede tener muchos usuarios.

La Clase caso se relaciona con la solicitud con una cardinalidad de uno a uno, lo que quiere decir que una solicitud tiene un único caso y un caso posee una única solicitud.

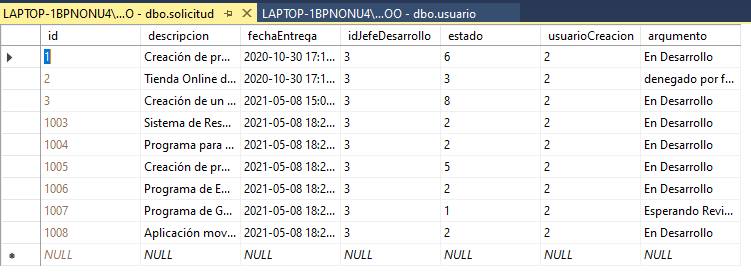
Entre la clase caso y bitácora hay una relación de composición, ya que la una bitácora no puede existir sin un caso

# Registros dentro de entidades

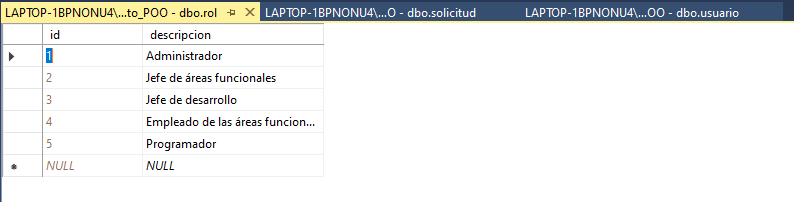
## Tabla Usuarios



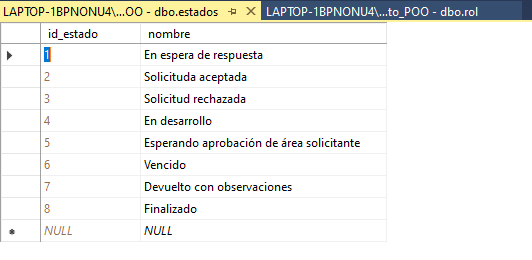
## Tabla Solicitudes



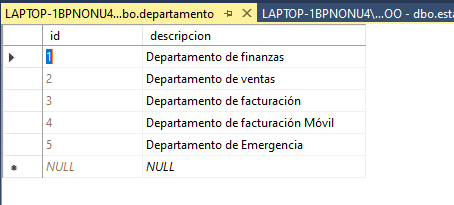
## Tabla Roles



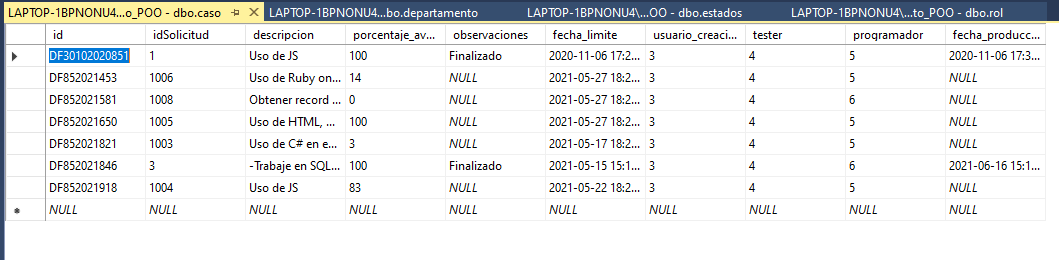
## Tabla Estados



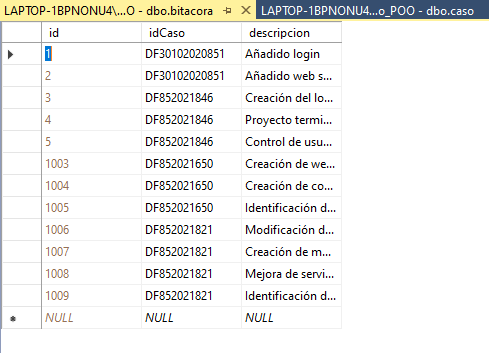
## Tabla Departamento



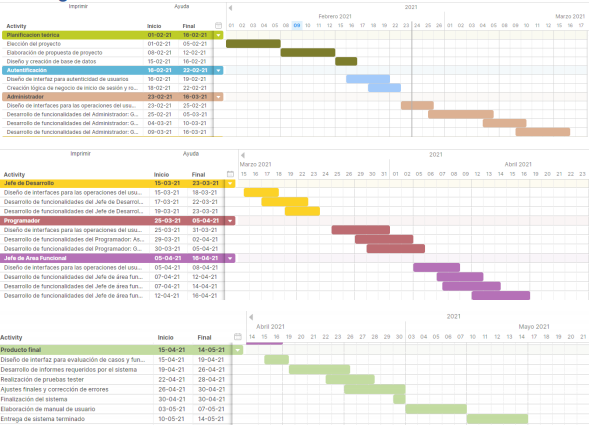
## Tabla Casos



## Tabla Bitácora



# Cronograma de actividades



# Conclusión

El programa desarrollado está orientado a mejorar la gestión de las necesidades de soluciones digitales dentro de una empresa sin importar su tamaño, ya que puede manejar una gran variedad de roles dentro del sistema, sin mencionar que se pueden agregar nuevos usuarios como sean requerido dentro de los departamentos, los cuales también pueden ser creados según las necesidades de cada empresa cruce.

La manera de como están divididas las responsabilidades dentro del sistema hace que el seguimiento de las solicitudes y casos sea más eficiente, ya que cada usuario se encarga de sus propias propuestas y como se van desarrollando hasta estar terminadas.

Nuestro programa está pensado para estar en una conexión LAN de una empresa con una estructura definida por el documento, donde detallan los roles y las acciones que cumple cada empleado. Con lo cual podemos decir que los roles e interacciones entre estos ya han sido definidos en documentos anteriores, por lo que adiciones de cualquier naturaleza, deben ser estudiadas y consideraras por los miembros del proyecto así también las revisiones a futuro están contempladas dentro del proyecto así proporcionando nuevas características o arreglos que se le puedan hacer al sistema cuando se ha implementado y se hagan pruebas de su rendimiento ya en su uso cotidiano.

# Bibliografía

A. Aho, J. H. (1987). *Data Structures and Algorithms.* Addison Wesley.

Gutiérrez, F. (2001). *Estructuras de Datos. Especificación, Diseño e Implementación.* Ed.Edicions UPC.

Joyanes Aguilar, L. (1987). *Metodología de la programación: diagramas de flujo algoritmos y programación estructurada.* México: McGraw Hill.

Joyanes Aguilar, L. (2003). *Fundamentos de programación : algoritmos y estructura de datos y objetos.* Madrid : McGraw-Hill.

Julien Bascha, L. J. (1999). Journal of Algorithms. Paderborn: Elsevier.

Loomis S, M. E. (1991). *Estructura de datos y organización de archivos.* México: Prentice-Hall Hispanoamericana.