UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS

Escuela de Ingeniería Informática

Ingeniería en Sistemas de Información

Programación II (IS-12)

Proyecto QA TRACKER

Michelle Monge Arguedas

Geancarlo Valverde Calderón

Profesor: Diego Vera Jiménez

San José, Costa Rica

Abril, 2014



Tabla de contenido

[Introducción 2](#_Toc385285986)

[Objetivo general 3](#_Toc385285987)

[Objetivos específicos 3](#_Toc385285988)

[Alcances 4](#_Toc385285989)

[Limitaciones 4](#_Toc385285990)

[Desarrollo 5](#_Toc385285991)

[Diagrama de clases 5](#_Toc385285992)

[Diagrama de base de datos 7](#_Toc385285993)

[Explicación en prosa de las principales funcionalidades del sistema 8](#_Toc385285994)

[Módulos desarrollados 9](#_Toc385285995)

[Mejoras 11](#_Toc385285996)

[Conclusiones 12](#_Toc385285997)

[Recomendaciones 13](#_Toc385285998)

[Glosario 14](#_Toc385285999)

[Bibliografía 15](#_Toc385286000)

# Introducción

En este apartado se brinda una breve explicación del proyecto asignado para la semana 14 del curso Programación II. Este informe tendrá diversas secciones como las descripciones de los problemas que deben resolverse, luego el objetivo general del trabajo y sus respectivos objetivos específicos.

Además se detallan los alcances del proyecto, las limitaciones u obstáculos que se debieron superar para realizar el trabajo, seguidamente, se explica el funcionamiento de cada una de las soluciones propuestas para cada problema.

Por último se anotan las conclusiones del desarrollo de la tarea, las recomendaciones para los estudiantes del curso de Programación II, el glosario con los términos utilizados para hacer esta asignación y las referencias empleadas para definir los conceptos.

# Objetivo general

Definir la solución más precisa para cumplir con los requerimientos del proyecto con base en los conocimientos adquiridos durante las lecciones e investigaciones por cuenta de los integrantes del equipo de trabajo, logrando el máximo desarrollo posible de la lógica y presentación de una excelente forma de satisfacer la necesidad del cliente, en este caso la elaboración de una herramienta que administre el proceso de desarrollo de software de la empresa, la aplicación se llamará “QATracker”.

# Objetivos específicos

Los objetivos específicos de esta asignación son: analizar el funcionamiento de un gestor de proyectos. Utilizar la lógica para implementar cada una de las funcionalidades especificadas, que permita realizar consultas y mantenimientos a los proyectos, usuarios, tareas e “issues” (problemas). Implementar el programa de tal forma que pueda ser accesado por los usuarios a través de una interfaz con un menú básico que incluya el inicio y cierre de sesión además de las opciones para realizar las consultas y mantenimientos descritos anteriormente.

# Alcances

Este informe pretende brindar la correspondiente explicación del código programado que dará solución al proyecto final planteado para los estudiantes del curso de Programación II. Se utilizó el software NetBeans IDE 7.4 para compilar y ejecutar el código, la herramienta Git para administrar las versiones del proyecto y el gestor de bases de datos MySQL para almacenar los datos de proyectos, usuarios, tareas y problemas “issues”.

# Limitaciones

Se debe destacar que para la realización no se encontraron limitaciones.

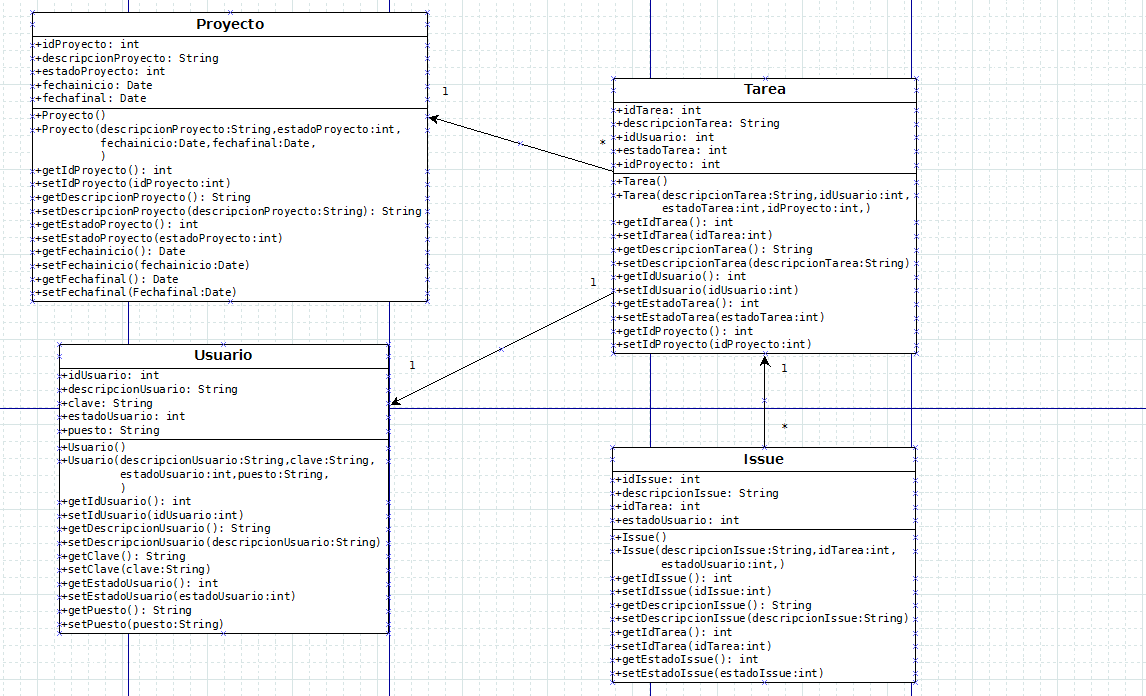
# Desarrollo

En este apartado se explicarán los diagramas de clases y de bases de datos, además se describen las principales funcionalidades, los módulos desarrollados, mejoras que se pueden realizar y otras observaciones.

## Diagrama de clases

A continuación se describe el diagrama de clases correspondiente al sistema diseñado en este proyecto.

Como se puede observar en la imagen, a un proyecto se le pueden asignar una o más tareas, cada tarea es asignada a un usuario y por último cada tarea puede tener uno o más defectos “issues”. Se pueden destacar dos relaciones de composición que son el objeto proyecto con el objeto tarea y éste con el objeto issues, además de la relación de asociación entre los objetos usuario y tarea.



## Diagrama de base de datos

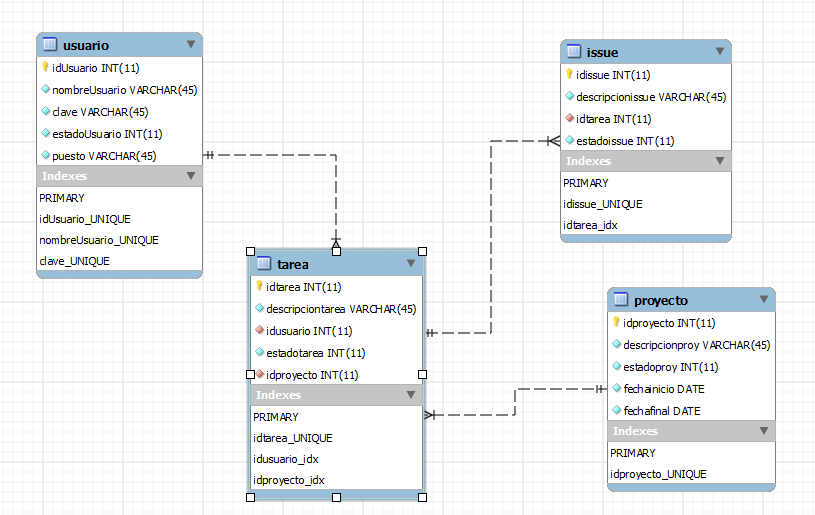
A continuación se describe el diagrama de bases de datos correspondiente al sistema diseñado en este proyecto. Como se puede observar se crearon 4 tablas con sus respectivos campos.

En el caso de la tabla proyecto, se estableció que el campo idproyecto almacenará un número único el cual servirá para diferenciar de un proyecto a otro. Además se almacenan otros datos como la descripción, estado, fecha de inicio y final de cada proyecto.

La tabla usuario, similar a la anterior tiene un campo que administre el registro único de id, además el campo nombreUsuario almacenará el nombre del colaborador, también se guarda la clave, el estado del usuario (activo o inactivo) y el puesto del empleado.

En el caso de la tabla tarea, se administra el identificador único de las tareas, su descripción, el usuario al que se le asignará la responsabilidad, el estado de la tarea (terminado, pendiente, reprocesando) y el proyecto al cuál estarán las tareas asociadas.

La tarea issue almacenará los problemas registrados por tarea, su descripción y el estado similar a la tabla tarea (terminado, pendiente, reprocesando).



## Explicación en prosa de las principales funcionalidades del sistema

A continuación se detalla una serie de características que ha permitido llevar a cabo la programación orientada a objetos en java del proyecto.

Un objeto en java puede ser cualquier objeto físico o conceptual de la vida real el cual está previsto de atributos y métodos. Una clase es un conjunto de métodos u objetos que comparten alguna similitud entre sí.

El proyecto llamado QA Tracker, es una herramienta para el departamento de control de calidad de la empresa IS-12. Esta permite hacer consultas a la base de datos, la cual despliega los proyectos, las tareas y los issues que existan. También permite el mantenimiento de estos, como lo es crear, borrar, actualizar, renombrar, todo dependiendo del caso.

Para la fácil navegación, la aplicación cuenta con un menú principal, esta posee cuatro opciones de navegación en el encabezado, las cuales son Inicio, Proyectos, Tareas e Issues. En la sección de Inicio, se cuenta con cuatro opciones diferentes: Log In, User, Sign Out y Exit. En las otras tres secciones faltantes se permite, ya sea hacer consultas o hacer mantenimientos, todo dependiendo de la sección.

En caso de que se desee realizar algún mantenimiento se debe de iniciar sesión en la sección de Inicio, de lo contrario solo se podrán hacer consultas generales. Cada ventana posee sus campos requeridos para su funcionamiento y está diseñada para que sea sencilla y fácil de trabajar para el usuario. La idea grafica promete un ambiente “User-Friendly” para procurar que no hayan conflictos al interactuar con la interfaz.

## Módulos desarrollados

Un método es un conjunto de sentencias que realizan una función específica dentro de la clase. Un evento constituye una forma en que la clase notifica a los usuarios de un objeto que algo sucede en el objeto, como, por ejemplo, que se ha hecho clic en un control de una interfaz gráfica de usuario.

Un mensaje es una instrucción que va dirigida a un objeto que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó. Una herencia es la propiedad que permite construirse a partir de otros objetos. Un atributo en java son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades.

En la opción de Log In en la sección de Inicio es donde se le permite al usuario iniciar sesión. Esto ya que, a pesar de que para realizar una consulta no se necesita logearse, para los mantenimientos se necesita ser un usuario activo de la empresa, por lo que es necesario identificarse como tal. Aquí se desplegara una ventana donde el usuario debe digitar su nombre de usuario y su contraseña, asi podrá darle mantenimiento a lo que desee.

Cabe recalcar que al iniciar sesión se desbloquearan dos opciones solo disponibles para usuarios activos. La primera es la sección de Settings en la opción User, en donde se podrán cambiar características del usuario actual. La segunda es que se desplegará en el encabezado de la aplicación el nombre del usuario que ha iniciado sesión y su puesto, esto para identificar.

En la opción de User se encuentran dos ítems: un ítem el cual puede ser accesado en cualquier momento, y otro que solo se desbloquea al iniciar sesión, el ítem Settings explicado anteriormente. El ítem fijo es la opción de New User, que permite crear un nuevo usuario. Cuando el usuario elije esa opción, se despliega una ventana donde se deberá de ingresar un nombre de usuario único, un puesto para el usuario y una contraseña, esto permitirá que el usuario creado pueda accesar a la aplicación e iniciar sesión.

La opción de Sign out solo tiene funcionalidad si actualmente se encuentra una sesión activa, ya que lo que esta hará es cerrar la sesión para permitir el ingreso de un nuevo usuario o el cierre seguro del sistema. Y en la última opción de la sección de Inicio, se encuentra el Exit, el cual solo se encarga de cerrar la aplicación.

Las opciones de consultas de Proyectos, Tareas e Issues, son libres de acceso, esto significa que no es necesario hacer un log in para consultar. En el caso de la consulta de proyectos, la ventana despliega una tabla con todos los proyectos existentes, así mismo posee un botón de buscar que permite buscar un proyecto en específico por medio de su id único. Para la consultar tareas, solo se despliegan las tareas asignadas a un proyecto en específico, por lo cual se debe de ingresar primero el id único del proyecto para poder desplegar las tareas de este en la tabla. De igual manera se permite la búsqueda de tareas específicas por medio de su número de id único. En la opción para consultar los issues, solo se despliegan los issues de una tarea seleccionada, para esto se ingresa el número de la tarea de la cual se desean consultar. Como en las consultas anteriores, también se permite la búsqueda de issues específicos por medio del número de id.

En la sección de mantenimiento se permite: crear, borrar y actualizar proyectos; crear, actualizar y renombrar tareas; crear y actualizar issues. En el mantenimiento de proyectos, en la ventana de crear, se le permite al usuario activo crear un nuevo proyecto por lo que debe ingresar la descripción, el estado (1.Entregado, 2.Pendiente, 3.Reprocesando), la fecha de inicio y la fecha final de este. La ventana actualizar permite cambiar la descripción, el estado, la fecha de inicio y la de final, de un proyecto ya creado anteriormente; y la ventana borrar, permite borrar un proyecto por medio del número de identificación de este.

En el mantenimiento de tareas, para la creación de una tarea, se debe ingresar en la ventana que se despliega, la descripción, el usuario al que se le asignara la tarea, el estado la tarea (con la misma referencia que los proyectos) y el proyecto al que se le asigna la tarea. En la actualización de tareas, se deben ingresar los mismos datos explicados en la oración anterior, para cambiar los datos anteriormente ingresados en la base de datos. Y para renombrar una tarea se debe ingresar la antigua descripción de esta y la que se desea ingresar en la ventana.

En el último mantenimiento disponible, el de issues, solo se permite crear y actualizar. Para crear se debe ingresar la descripción, la tarea a la que se le agrega el issue y el estado, de igual manera para actualizar.

## Mejoras

En este apartado se describen las posibles mejoras a realizar al proyecto. Una aplicación de este calibre podría ser implementada en ambiente web, esto facilitaría el ingreso más abierto a los usuarios de la aplicación.

En el campo de los usuarios se podría implementar un espacio donde no se pueda tener al mismo usuario como conectado dentro de dos instancias diferentes, es decir que no hayan más de una sesión abierta por usuario, como se puede notar esta característica adicional está muy relacionada con el tema de accesar la aplicación desde un navegador web.

# Conclusiones

La comprensión de la programación orientada a objetos con énfasis en el lenguaje Java (según el curso matriculado) permitió que la propuesta aquí descrita se lograra realizar exitosamente. El análisis de los problemas antes de empezar a programar, fue elemento clave para no sufrir en exceso para brindar la mejor solución posible, apoyados en la investigación y en el método de ensayo y error.

La implementación del proyecto en ambiente gráfico, fue un gran reto para el equipo de trabajo porque aún con la investigación realizada no se lograba entender por qué no funcionaba la aplicación, sin embargo, después de varios intentos se encontró un detalle en la ubicación de las librerías utilizadas para la interfaz.

# Recomendaciones

Es muy importante comentar el código para que en las ocasiones donde el programa contenga errores, se puedan ubicar los mismos con mayor facilidad. Además se debe resaltar la importancia de utilizar herramientas para administrar las versiones del proyecto, en este caso se utilizó el programa GitHub, el cual facilitó el avance del proyecto evitando que los miembros del equipo utilizaran su tiempo trabajando en archivos distintos.

Otro aspecto a rescatar y que está relacionado al punto anterior es probar el código según los avances obtenidos y no hasta el final, porque podría ser que la lógica implementada no sea la correcta.

El trabajo en equipo es un elemento con el que se pudo contar para la realización del proyecto, no es lo mismo desarrollar una solución de manera individual que con dos mentes frescas que puedan abarcar una investigación mayor y una mejor depuración de la solución que se está brindando.

# Glosario

En este apartado se describen conceptos importantes para que el lector pueda comprender el informe de mejor forma.

Atributos: son las características propias del elemento a programar, en este caso persona y propiedad.

Clase: es un tipo de dato definido por el usuario.

Diagrama de bases de datos: es una representación de las tablas utilizadas para la implementación del proyecto.

Diagrama de clases: es una representación de los atributos que tienen los objetos o bien, relaciones con otros objetos.

Import: es utilizado para poder emplear las librerías necesarias para trabajar con la clase Scanner por ejemplo, que captura los datos ingresados desde el teclado.

Issue: sinónimo de defecto, se relacionan con las tareas cuando estas tienen problemas que impidan que las mismas se entreguen en el tiempo estimado.

Métodos: son funciones definidas dentro de una clase que pueden modificar los datos privados de una clase.

Objeto: es la instancia de un objeto, por el cual se puede hacer uso de una clase para el fin requerido, en el proyecto se definieron objetos usuarios, tareas, proyectos e “issues”.

Programación por capas: es la arquitectura de software que consiste en dividir la lógica de la aplicación del negocio de la lógica de datos y de la presentación de la información del usuario. Esto permite que el equipo de trabajo se pueda dividir para atender las diferentes capas de forma más fácil y además el mantenimiento de las aplicaciones solo se realizaría en la(s) capa(s) que así sean requeridas.

# Bibliografía

En este apartado se citan las fuentes de información utilizadas para implementar y documentar el proyecto.

Usuario MADRUGADOR2010, “Poner fondo a un JDesktopPane en Java con MADRUGADOR”, video consultado en la red social Youtube el 13 de abril, https://www.youtube.com/watch?v=vrNc5bBbJ3g