

Algoritmos y Estructuras de Datos

Tarea Programada # 2

**Documentación “ClientesBac”**

Luis Diego Delgado Vásquez (2014119779)

Wayner Stanley Valverde Monge (2014004869)

Semestre II

2014



Contenido

[Descripción del problema 3](#_Toc401032875)

[Diseño del programa 3](#_Toc401032876)

[Librerías 5](#_Toc401032877)

[Análisis de resultados 5](#_Toc401032878)

[Manual de Usuario 6](#_Toc401032879)

[Instrucciones de Uso: 10](#_Toc401032880)

[Conclusiones Personales: 14](#_Toc401032881)

# Descripción del problema

Solventar las carencias de conocimiento en el lenguaje de programación Java y sus respectivas formas de compilación y utilización, al igual que experimentar con el uso de estructuras de datos para distintas aplicaciones, mediante el desarrollo de un programa para la gestión de atención a clientes en un banco, implementando colas de prioridad, asignación de cajas, envío de correos y reportes.

# Diseño del programa

**Decisiones de diseño:** El programa fue diseñado pensando primeramente en todos los requerimientos que este iba a tener, considerando así ya otros programas existentes en la industria del software comercial con una funcionalidad parecida, determinando así las pantallas que iban a ser necesarias y las funcionalidades lógicas (estructuras de datos, funciones, procedimientos, etc) que luego se irían amoldando al objetivo deseado para la aplicación.

Finalmente determinando el programa en 8 pantallas, la primera como pantalla principal mostrando las opciones del programa, una pantalla de configuración para determinar el logo y la cantidad de cajas del banco, una pantalla para poder solicitar un tiquete mediante el ingreso de datos personales, una pantalla para liberar las cajas una vez que ya no están en uso, otra que genera los reportes en la tabla y 2 pantallas auxiliares con la muestra de los gráficos.

**Algoritmos usados:**

1. Determinar los requerimientos

2. Analizar de los requerimientos

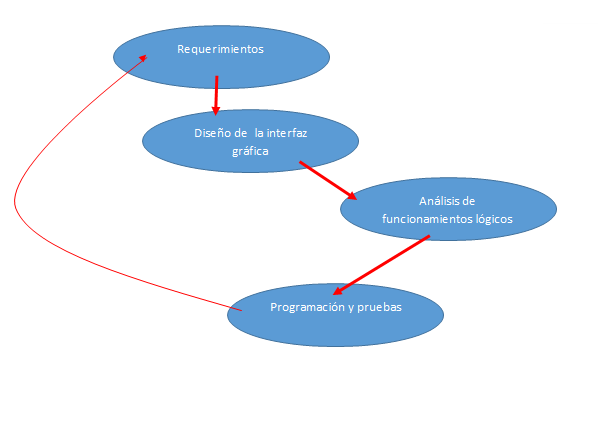
3. Diseñar la interfaz gráfica

4. Programar la interfaz gráfica:

* Desarrollo de Colas de prioridad
* Creación de colas de tipo caja
* Algoritmos de ordenamiento
* Envío de correos
* Creación de gráficos

5. Probar la validez y factibilidad de uso para el usuario.

**Diagramas Lógicos:**



# Librerías

**Substance-4.3:** Se trata de una librería orientada a la interfaz gráfica, ya que posee bastantes “skins” para cambiar el aspecto del proyecto

**Uso en el proyecto:** Se utilizó esta librería para lograr que la interfaz del programa fuera más personalizada, y un poco más amigable con el usuario.

**Jfreechart-1.0.19:** Librería especializada en creación de gráficos.

**Uso en el proyecto:** Fue la librería que hizo posible mostrar los gráficos solicitados en la pantalla de reportes, tanto de barras como pastel.

**Javax.mail:** Logra manejo de correos, envío mediante Gmail.

**Uso en el proyecto:** Se implementó en el envío de notificaciones a los clientes.

# Análisis de resultados

**Objetivos Alcanzados:**

* *Envío de correos:* El programa logra enviar correos de manera efectiva a sus clientes
* *Creación de cajas:* Se logra crear un conjunto de cajas de manera dinámica.
* *Creación de colas de prioridad:* El programa crea y ordena los elementos de una cola bajo un criterio de prioridad previamente asignado.
* *Asignar turnos:* La asignación de turnos se da de manera eficiente, basándose en la cola y en las cajas vacías
* *Creación de gráficos:* Se investigó e implemento el método correcto para lograr representar gráficos con la información o datos de un tiempo determinado.

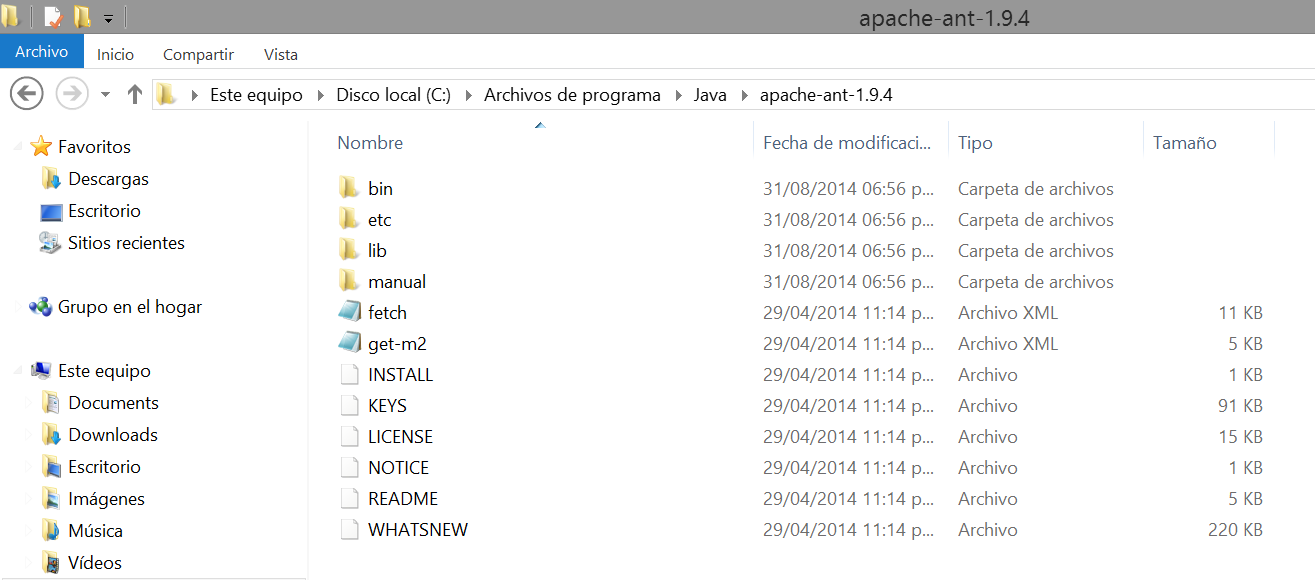
**Objetivos no alcanzados:**

* *Finalizar el proyecto con un poco más de tiempo:* El objetivo no se alcanzó debido a que la carga académica está resultando ser muy pesada, inclusive en el fin del semestre.
* *Manejo eficiente de los punteros:* Las referencias nulas de los punteros fue un problema que estuvo presente a lo largo del desarrollo del proyecto.
* *Tabla dinámica:*No se logró implementar los algoritmos de ordenamiento de una manera precisa, debido a que se invirtió mucho tiempo tratando de solucionar problemas.

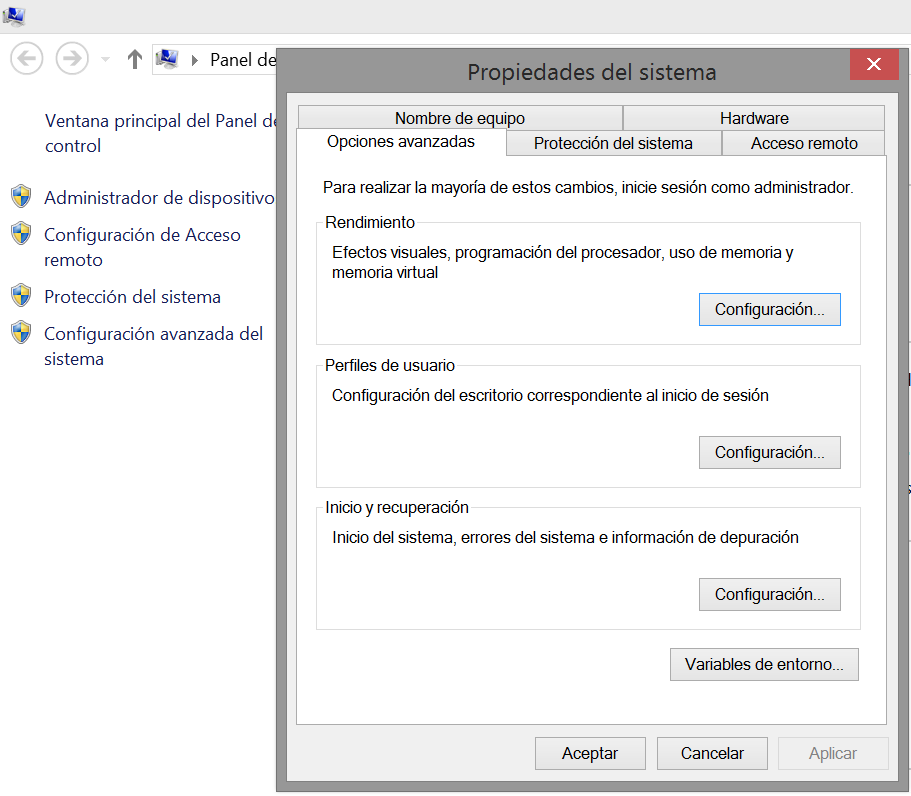
# Manual de Usuario

***Instrucciones de compilación y ejecución***

1. Descargar el antApache “binary distributions” de <http://ant.apache.org/>
2. Descomprimirlo en una ubicación fija dentro de la computadora

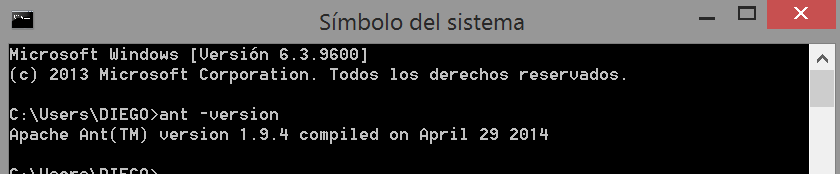


1. Ir a configuración avanzada del sistema por medio del panel de control o propiedades del sistema y seleccionar variables de entorno

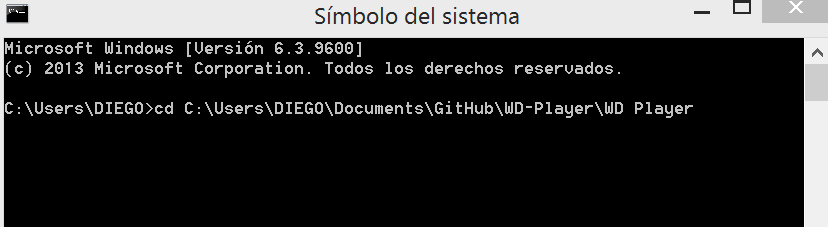


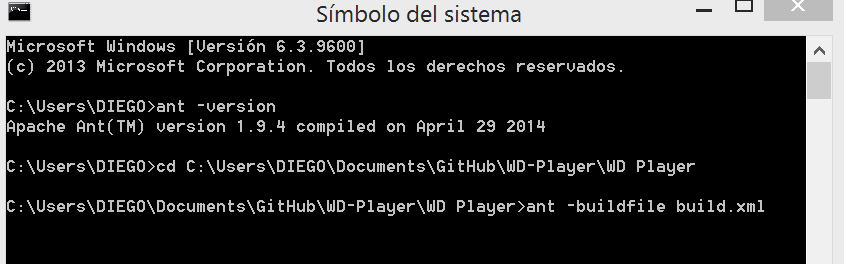
1. Crear 2 variables: primera llamada ANT\_HOME y pegarle la ubicación de donde esta descomprimido el ANT, segunda llamada JAVA\_HOME y pegarle la ubicación del jdk ene l sistema.

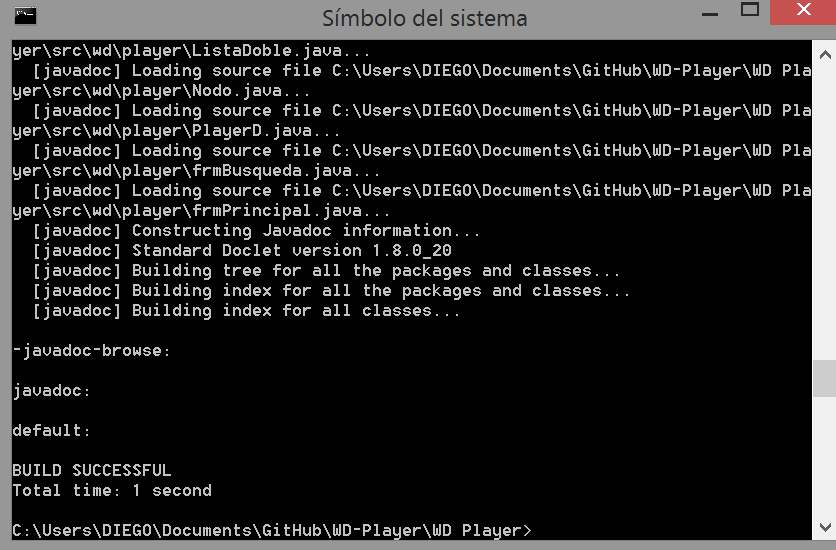
Edita Patch y pegarle al principio la ubicación de la carpeta bin dentro de la Carpeta del ANT.

1. Abrir la consola (escribir cmd en el inicio)
2. Comprobar que el ANT se instaló correctamente digitando en el cmd “ant -version” y aparezca algo parecido a lo de la imagen

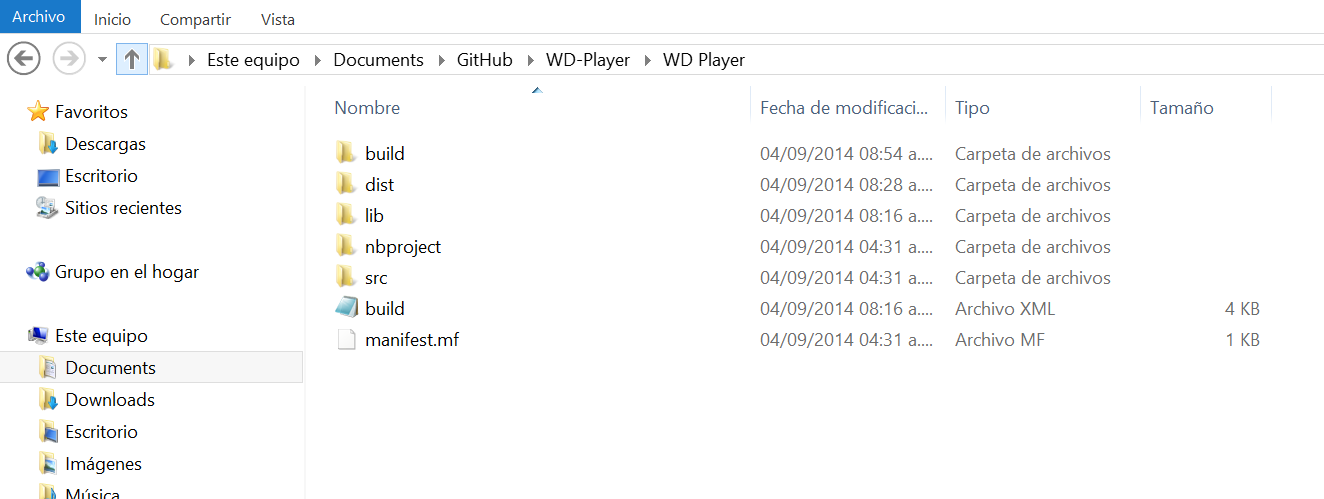
7. Abrir la ruta del proyecto donde se encuentra el build.xml (por defecto) el cual contiene la información necesaria para la compilación en ANT por medio de “cd ruta”.



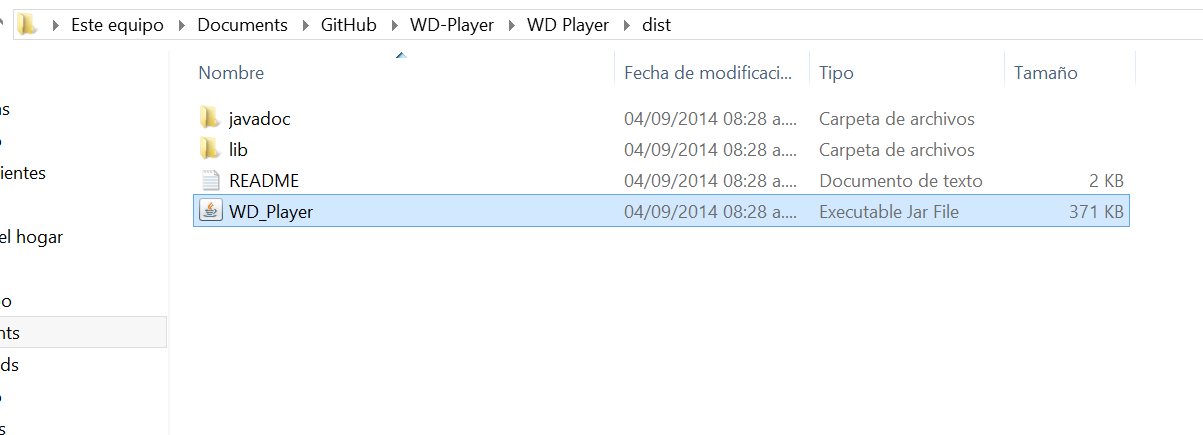
8. Por último ejecutar “ant –buildfile build.xml” donde build.xml corresponde al archivo xml y por último deberíamos ver un mensaje que diga “build successful” como en la segunda imagen y ya habremos compilado nuestro proyecto con ANT.



9. Ir a la carpeta del proyecto y buscar la carpeta llamada dist, esa es la que habrá creado ANT y donde se encuentra el ejecutable de la aplicación



10. Finalmente ejecutar el archivo .jar y listo!

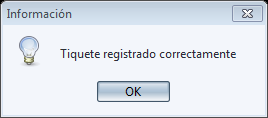


# Instrucciones de Uso:

Se presenta el menú principal:

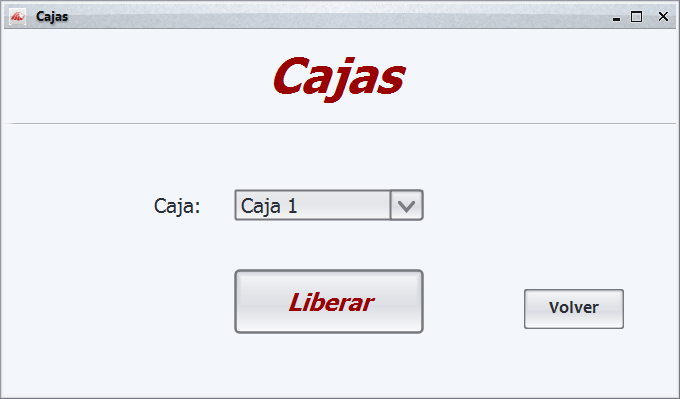
**Crear tiquetes:**

Se accede a la pantalla al presionar el botón “Tiquetes, en dicha pantalla se procede a escoger el tipo de cliente mediante un comboBox, y se llena el nombre y el correo, posteriormente presiona “Procesar” y se recibe un mensaje emergente.



**Liberar cajas:**

Se accede a esta pantalla el presionar el botón “Liberar Caja”, se escoge cuál de las cajas se va a liberar, mediante un comboBox, seguidamente se presiona el botón de “Liberar”, para poner a disposición la caja liberada.



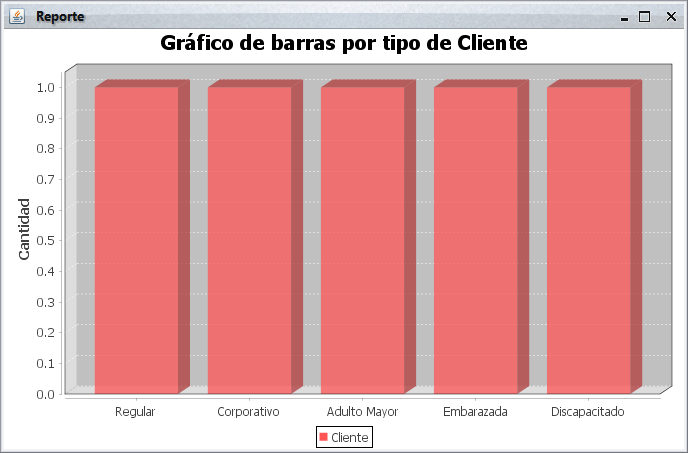
**Reportes:**

Se presiona el botón “Reportes” en el menú principal, en la pantalla siguiente “Enjoy”, aparecerá la tabla con información de los clientes que puede ser filtrada mediante las fechas, además botones al costado superior derecho para ver los gráficos de barra y pastel.



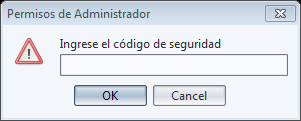


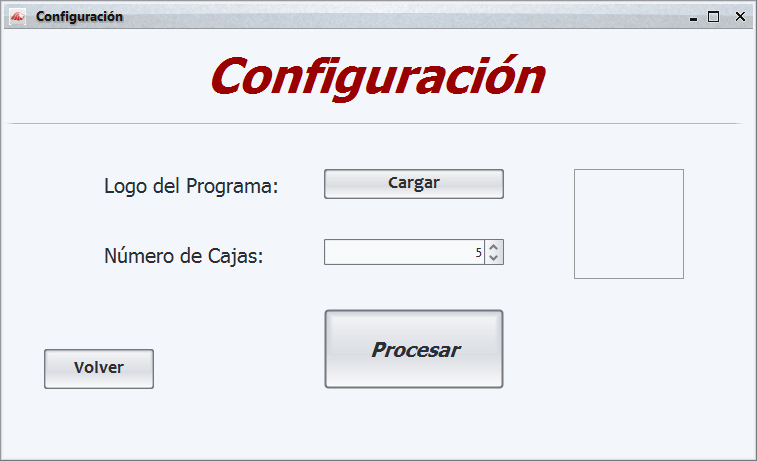




**Configuración:**

En el menú principal se accede a una de las opciones de la barra de tareas, para configurar el programa, aparecerá una ventana que pide una contraseña, la cual es “topsecret”, de esta manera se ingresa a una pantalla con las opciones de cargar un logo para el programa, y definir la cantidad de cajas, el botón de procesar guardará los cambios.





# Conclusiones Personales:

La implementación del programa en papel suena bastante fácil, útil y realista, pero al empezar a desarrollar las estructuras de datos necesarias, su nivel de complejidad aumenta, hay que seguir mejorando en las técnicas de desarrollo de estas estructuras ya que se experimentó muchos problemas a la hora de cambiar punteros, pero aun así eran un tipo de error difícil de percibir ya que la lógica del código parece estar bien en todo momento, pero no funciona como se espera.

También se puede tomar el lado positivo del alto nivel de aprendizaje que se obtuvo al conocer acerca de gráficos y correos en Java, más muchos métodos realizados que ayudan a mejorar nuestra lógica como programadores.