

SoftBank

Especificación de Requisitos Software

Autores

- Ignacio García Sánchez-Migallón
- David Gorricho San Juan
- Carlos Gavidia Ortiz
- Lorenzo José de la Paz Suárez
- Iván Monterrubio Cerezo
- Diego Sánchez Muniesa

Control de cambios

Número de versión	Fecha	Autores	Descripción
V.0.7	16/12/2015	Iván, Carlos, David, Lorenzo, Ignacio, Diego	Planificación y distribución del trabajo. Gran desarrollo de la especificación de requisitos
V.0.8	18/12/2015	Iván, Carlos, David, Lorenzo, Ignacio, Diego	Descripción y desarrollo de la mayor parte de las funciones
V.0.85	21/12/2015	David, Lorenzo	Finalización del resto de funciones inacabadas (las relacionadas con las acciones)
V.0.9	23/12/2015	Ignacio, Diego	Corrección de errores con varias modificaciones de errores detectados
V.1.0	26/12/2015	Iván, Carlos	Revisión completa y exhaustiva de la especificación de requisitos

Índice

1.	Introducción	1
1.1	Propósito	1
1.2	Alcance	1
1.3	Definiciones, acrónimos y abreviaturas	2
1.4	Referencias.....	2
1.5	Resumen.....	2
2.	Descripción general.....	3
2.1	Perspectiva del producto	3
2.2	Funciones del producto.....	3
2.3	Características del usuario	5
2.4	Restricciones	5
2.5	Supuestos y dependencias.....	6
2.6	Requisitos futuros	6
3.	Requisitos específicos	7
3.1	Interfaces externos.....	7
3.2	Requisitos funcionales.....	7
3.3	Requisitos de rendimiento	17
3.4	Requisitos sobre la persistencia de datos.	17
3.5	Restricciones de diseño.....	17
3.6	Atributos del sistema software	18

1. Introducción

1.1 Propósito

El propósito de este documento es realizar la especificación de los requisitos. Partimos de la base del funcionamiento del software a diseñar para realizar una planificación, una estimación de costes y prever el esfuerzo de su desarrollo. En las páginas siguientes identificaremos la funcionalidad de la aplicación así como las diferentes interfaces externas, atributos y restricciones del diseño impuestas de cara a su implementación.

Los usuarios a los que este documento va dirigido son personas con conocimientos de Informática y diseño de Software, así como posibles clientes que quiera obtener una información extra del proyecto.

1.2 Alcance

Nombre del producto: *“SoftBank”*

El objetivo de este Software es la gestión del funcionamiento de un banco así como de los trámites y gestiones típicas del día a día en una empresa de tales características. Opera los datos personales de los usuarios y el flujo de capital de las mismas desde el punto de vista del operario (Transferencias, creación de cuentas, pago de facturas, gestión de acciones...)

El acceso a algunas características es restringido, lo que implica que en función del cargo del operario que haga uso de la misma el número de acciones disponibles se encontrará limitado de un modo u otro.

- Director: Acceso completo.
- Gerente: Encargado de los ingresos y retiradas de dinero por parte del usuario y actualización de Cartillas.
- Bróker: Gestión de compra y venta de acciones
- Gestor: Da consejos sobre la utilización de los activos disponibles de cada usuario.
- Prestamista: Decide sobre la autorización de créditos, toma en cuenta la cantidad de fondos en la cuenta así como los ingresos.

Los beneficios derivados de la utilización de nuestro producto son claros: toda gestión se realiza de forma fácil y rápida, además de eficiente al ser realizada por expertos en la materia. Por otro lado es segura debido a la necesidad de utilizar las contraseñas de los operarios en todo momento para la realización de cualquier tipo de acción. Por otro lado guarda registro de toda acción realizada dando lugar a una base de datos de acciones realizadas para evitar malentendidos con los clientes.

1.3 Definiciones, acrónimos y abreviaturas

Préstamo: Cantidad de dinero que se da a una persona para que la use durante un tiempo y después la devuelva.

1.4 Referencias

No hay referencias a documentos externos.

1.5 Resumen

El documento está dividido en tres partes. La primera parte explica la funcionalidad del proyecto así como da una breve idea de su funcionamiento. La segunda parte se encarga de identificar las funciones que debe tener el producto así como sus restricciones y posibles requisitos futuros. La tercera y última parte analiza de forma completa y exhaustiva los requisitos dividiendo las diferentes funciones.

En los apéndices se encuentran prototipos y versiones tempranas de las interfaces externas .

2. Descripción general

2.1 Perspectiva del producto

El producto se ejecuta en los ordenadores del banco, sin necesidad de instalación.

No se necesita conexión a internet para su correcto funcionamiento pues el ámbito de la aplicación es puramente local por motivos de seguridad.

La aplicación hace uso de numerosos métodos de autenticación para identificar el operario que se encuentra haciendo uso de la misma y el acceso a las diferentes acciones disponibles a cada uno.

2.2 Funciones del producto

A continuación se encuentra una lista con las diferentes funcionalidades del producto así como una breve explicación de las mismas.

Creación de una cuenta de usuario

- Creación de contraseña y número de cuenta a un nuevo usuario.

Creación de una cuenta de empleado

- Asignación de contraseña al empleado de un puesto específico.

Comprobar saldo de una cuenta

- Ofrece información sobre la cantidad de dinero disponible.

Retirar capital

- Permite retirar dinero de la cuenta bancaria introducida

Ingresar capital

- Permite ingresar dinero en la cuenta bancaria introducida

Suspensión de una cuenta de usuario

- Elimina la cuenta de un usuario

Transferencia bancaria

- Permite traspasar fondos de una cuenta a otra

- ✚ Suspensión de una cuenta de empleado
 - Elimina la cuenta de un empleado (Despido)
- ✚ Comprobación de identificación de usuario
 - Comprueba que son correctas las credenciales del usuario
- ✚ Comprobación de identificación de empleado
 - Comprueba que es correcta la credencial ofrecida por el empleado
- ✚ Comprobación de contraseña/PIN
 - Comprueba que la contraseña introducida es correcta
- ✚ Solicitud de un préstamo
 - Permite al usuario pedir un préstamo
- ✚ Comprobación de la viabilidad de un préstamo
 - Evalúa si el préstamo es concedido al usuario
- ✚ Cálculo del interés de un préstamo
 - Calcula mediante una función el valor del interés del préstamo
- ✚ Compra de acciones
 - Permite al usuario la compra de acciones de diferentes compañías
- ✚ Venta de acciones
 - Permite al usuario la venta de acciones de diferentes compañías
- ✚ Asesoramiento sobre acciones
 - Da consejos sobre la compra/venta de diferentes acciones
- ✚ Solicitud de hipoteca
 - Da la oportunidad de solicitar una hipoteca al usuario
- ✚ Comprobación de la viabilidad de una hipoteca
 - Evalúa si la hipoteca es concedida al usuario

Asesoramiento bancario general

- Ofrece consejo al usuario sobre las acciones que puede realizar con su cuenta

Actualización de una cartilla

- Actualiza la cartilla del usuario con toda la información ocurrida hasta el momento de la actualización.

2.3 Características del usuario

El usuario final, es decir, el que hará uso del programa será una persona profesional, con conocimiento de causa y cierta experiencia en la utilización de sistemas informáticos de gestión así como conocimientos de economía.

El objetivo del producto es que sea sencillo de utilizar incluso para el que no posea ningún tipo de conocimiento previo sobre lo anterior.

2.4 Restricciones

Las restricciones en este proyecto son las siguientes:

- ❖ La aplicación debe ser de Escritorio
- ❖ El lenguaje de implementación ha de ser Java
- ❖ La persistencia de los datos debe hacerse en formato texto o en formato relacional.
- ❖ Debe haber polimorfismo
- ❖ Un tipo de empleado no podrá realizar las acciones de otro bajo ningún concepto
- ❖ Los empleados no deben poder tener acceso a la contraseña de los usuarios
- ❖ No puede haber saldos negativos en las cuentas
- ❖ La interfaz ha de ser simple e intuitiva, de modo que los empleados, sin formación previa en el uso de la aplicación, sean capaz de usarla.
- ❖ La aplicación debe ser segura, las contraseñas de los usuarios nunca son vistas y las de los empleados solo en el momento de su creación.

- ❖ No puede tener acceso a internet y viceversa para evitar vulneraciones en la seguridad.

2.5 Supuestos y dependencias

La aplicación es dependiente de la máquina virtual de Java, es decir, es necesaria su previa instalación para poder hacer uso de la misma, de otro modo, no es posible su utilización.

2.6 Requisitos futuros

Posibles ampliaciones de la funcionalidad podrían ser los Programas de Puntos y Fidelidad para premiar con regalos a los clientes más activos.

Otra posible ampliación de la funcionalidad sería la creación de planes de pensiones y su gestión así como consejos sobre el tipo más recomendado para el usuario.

3. Requisitos específicos

3.1 Interfaces externos

Disponemos de dos interfaces hardware que interaccionan con la aplicación. Una de ellas es el lector de tarjetas de crédito, que nos facilitará la información de la cuenta del cliente. La otra se encarga de leer y escribir en la cartilla de los clientes con la intención de proceder a su actualización, accediendo para ello a la base de datos de la aplicación.

3.2 Requisitos funcionales.

Creación de una cuenta de cliente

- **Prioridad**: alta
- **Estabilidad**: media
- **Descripción**: solicita los datos personales del cliente (nombre completo, NIF, domicilio, etc.) y crea una cuenta de usuarios en la base de datos de la aplicación, generando una contraseña aleatoria que servirá particularmente para este cliente
- **Entrada**: datos personales del cliente
- **Salida**: contraseña de 4 dígitos aleatoria
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: base de datos de usuarios
- **Acción**: crear un cuenta
- **Precondición**: el cliente no puede tener otra cuenta ya creada
- **Postcondición**: la cuenta ha sido creada
- **Efectos laterales**: se guarda la cuenta en la base de datos

Creación de una cuenta de empleado

- **Prioridad**: muy alta
- **Estabilidad**: media
- **Descripción**: solicita los datos personales del el empleado (nombre completo, NIF, domicilio, etc.) , creara un a nueva cuenta de empleado, diferenciando su cargo en la empresa, ya que dependiendo de esta podrá realizar diferentes acciones en la aplicación

- **Entrada**: datos personales del empleado y su cargo
- **Salida**: contraseña de 4 dígitos aleatoria
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: base de datos de los empleados en la empresa
- **Acción**: crear un cuenta de empleado
- **Precondición**: el empleado no puede tener otra cuenta ya creado diferente cargo en la empresa
- **Postcondición**: la cuenta ha sido creada
- **Efectos laterales**: se guarda la cuenta en la base de datos de los empleado

Comprobar saldo de una cuenta

- **Prioridad**: alta
- **Estabilidad**: media
- **Descripción**: Dado una cuenta de un cliente registrado en la base de datos de la aplicación(comprobado que exista)
- **Entrada**: usuario y contraseña del cliente
- **Salida**: Saldo del cliente en su cartilla.
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: base de datos del cliente
- **Acción**: mostrar el saldo del cliente
- **Precondición**: Dar una contraseña y un usuario que existan en la aplicación
- **Postcondición**: mostrar su saldo
- **Efectos laterales**: mostrar el salgo de un cliente por pantalla

Retirar capital

- **Prioridad**: muy alta
- **Estabilidad**: alta
- **Descripción**: retira el dinero que solicite el cliente de su cuenta
- **Entrada**: cantidad de dinero que haya solicitado el usuario
- **Salida**: cantidad de dinero retirado de la cuenta
- **Origen**: petición del usuario

- **Destino**: sistema
- **Necesita**: cantidad de dinero a retirar
- **Acción**: mostrar la cantidad de dinero a retirar
- **Precondición**: la cantidad de dinero a retirar sea menor o igual al dinero que haya en la cuenta
- **Postcondición**: mostrar la cantidad de dinero que haya retirado
- **Efectos laterales**: mostrar la retirada de dinero

Ingresar capital

- **Prioridad**: muy alta
- **Estabilidad**: alta
- **Descripción**: ingresar el dinero que solicite el cliente de su cuenta
- **Entrada**: cantidad de dinero que haya solicitado el usuario
- **Salida**: cantidad de dinero que tenga en su cuenta
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: cantidad de dinero a ingresar
- **Acción**: mostrar la cantidad de dinero ingresado
- **Precondición**: ninguna
- **Postcondición**: mostrar la cantidad de dinero que haya ingresado
- **Efectos laterales**: mostrar el ingreso de dinero

Suspensión de una cuenta de usuario

- **Prioridad**: alta
- **Estabilidad**: baja
- **Descripción**: elimina de la base de datos la cuenta de un cliente existente
- **Entrada**: cuenta del cliente a eliminar
- **Salida**: indicación de si la cuenta ha sido eliminada correctamente
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: base de datos de usuarios
- **Acción**: eliminar un cuenta
- **Precondición**: el cliente debe tener una cuenta ya creada

- **Postcondición**: la cuenta ha sido eliminada
- **Efectos laterales**: se borra la cuenta en la base de datos y se recoloca la misma

Transferencia bancaria

- **Prioridad**: alta
- **Estabilidad**: media-baja
- **Descripción**: traspaso que un cliente puede hacer de parte del capital de su cuenta a la cuenta de otro cliente diferente
- **Entrada**: cantidad del dinero a traspasar y cuenta del cliente que recibe el dinero
- **Salida**: indicación de si la transferencia ha sido correcta
- **Origen**: cuenta del cliente que realiza el traspaso
- **Destino**: cuenta del cliente que recibe el traspaso
- **Necesita**: cuenta de los dos clientes que realizan la operación
- **Acción**: disminución del capital del cliente que realiza el traspaso y aumento del capital en la misma cantidad del cliente que lo recibe
- **Precondición**: el cliente que hace la transferencia debe tener una cuenta existente y el suficiente capital disponible
- **Postcondición**: el cliente que recibe la transferencia debe tener una cuenta existente
- **Efectos laterales**: el capital del cliente que traspasa y el que recibe, disminuye y aumenta, respectivamente, en la misma cantidad

Suspensión de una cuenta de empleado

- **Prioridad**: alta
- **Estabilidad**: baja
- **Descripción**: elimina de la base de datos la cuenta de un empleado existente
- **Entrada**: cuenta del empleado a eliminar
- **Salida**: indicación de si la cuenta ha sido eliminada correctamente
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: base de datos de empleados
- **Acción**: eliminar un cuenta
- **Precondición**: el empleado debe tener una cuenta ya creada

- **Postcondición**: la cuenta ha sido eliminada
- **Efectos laterales**: se borra la cuenta de la base de datos y se recoloca la misma

Comprobación de identificación de usuario

- **Prioridad**: alta
- **Estabilidad**: alta
- **Descripción**: recibe el DNI del usuario y su PIN y comprueba si existe el usuario y si se corresponde con el PIN almacenado. Nos devuelve el lugar del array en el que se encuentra el usuario o -1 si no se encuentra disponible.
- **Entrada**: DNI y PIN
- **Salida**: posición del array en el que se encuentra el usuario.
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: fichero de usuarios
- **Acción**: comprobar que el usuario es correcto.
- **Precondición**: la lista de usuarios no se encuentre vacía
- **Postcondición**: el usuario (si se encuentra en la lista) se debe corresponder con su PIN. Si no se devolverá -1 y se indicará que el nombre de usuario/PIN no es correcto.
- **Efectos laterales**: no tiene

Comprobación de identificación de empleado

- **Prioridad**: alta
- **Estabilidad**: alta
- **Descripción**: recibe el DNI del empleado y su contraseña y comprueba si existe el empleado y si se corresponde con su contraseña almacenada. Nos devuelve el lugar del array en el que se encuentra el empleado o -1 si no se encuentra disponible.
- **Entrada**: DNI y contraseña
- **Salida**: posición del array en el que se encuentra el empleado.
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: fichero de empleados

- **Acción**: comprobar que el empleado se encuentra registrado en el sistema.
- **Precondición**: la lista de empleados no se encuentre vacía y sea accesible
- **Postcondición**: el usuario (si se encuentra en la lista) se debe corresponder con su contraseña. Si no se devolverá -1 y se indicará que el nombre de usuario/contraseña no es correcto.
- **Efectos laterales**: no tiene

Comprobación de contraseña/PIN

- **Prioridad**: alta
- **Estabilidad**: baja
- **Descripción**: comprueba que el usuario/empleado suministrado se corresponde con la PIN/ contraseña proporcionada
- **Entrada**: usuario/empleado y PIN/contraseña
- **Salida**: devuelve si se corresponde la contraseña o no
- **Origen**: petición del sistema
- **Destino**: sistema
- **Necesita**: fichero de usuarios y empleados
- **Acción**: comprobar contraseña
- **Precondición**: que exista el usuario/empleado
- **Postcondición**: ninguna
- **Efectos laterales**: ninguno

Solicitud de un préstamo

- **Prioridad**: alta
- **Estabilidad**: baja
- **Descripción**: petición por parte de un usuario del banco de una cantidad determinada de dinero a pagar en un tiempo X y con un interés determinado
- **Entrada**: petición del usuario del banco
- **Salida**: llamada a la siguiente función (Comprobación viabilidad de un préstamo)
- **Origen**: petición del usuario
- **Destino**: sistema
- **Necesita**: base de datos de usuarios

- **Acción:** Solicitar préstamo
- **Precondición:** el cliente debe tener una fuente de ingresos y una cuenta
- **Postcondición:** no tiene
- **Efectos laterales:** se llama a la siguiente función

✚ Comprobación de la viabilidad de un préstamo

- **Prioridad:** alta
- **Estabilidad:** baja
- **Descripción:** analiza el capital del usuario y su fuente de ingresos
- **Entrada:** solicitud del préstamo
- **Salida:** aceptación/denegación del préstamo
- **Origen:** petición del usuario
- **Destino:** sistema
- **Necesita:** base de datos de usuarios
- **Acción:** concesión de un préstamo
- **Precondición:** el cliente debe tener fuente de ingresos
- **Postcondición:** la necesidad de pagarlo
- **Efectos laterales:** se guarda la cuenta en la base de datos

✚ Cálculo del interés de un préstamo

- **Prioridad:** alta
- **Estabilidad:** alta
- **Descripción:** en función de los ingresos del usuario y la cantidad prestada se genera un interés que irá aumentando el dinero a devolver progresivamente
- **Entrada:** préstamo solicitado
- **Salida:** porcentaje de interés
- **Origen:** concesión de un préstamo
- **Destino:** sistema
- **Necesita:** base de datos de usuarios
- **Acción:** cálculo de interés
- **Precondición:** el préstamo debe haber sido concedido
- **Postcondición:** el cliente debe aceptarlo y pagarlo
- **Efectos laterales:** se guarda la cuenta en la base de datos

Compra de acciones

- **Prioridad**: media
- **Estabilidad**: media
- **Descripción**: el cliente solicita la compra de acciones, por lo que se lo comunica al empleado 'Broker' del SoftBank. El 'Broker' se introduce en el mercado de valores de la Bolsa de Madrid y compra las acciones que desea el cliente.
- **Entrada**: capital dispuesto por el cliente para la compra.
- **Salida**: acciones compradas.
- **Origen**: petición del usuario.
- **Destino**: el banco SoftBank.
- **Necesita**: capital
- **Acción**: compra de valores/acciones
- **Precondición**: el cliente debe de tener capital suficiente.
- **Postcondición**: ha habido una compra de acciones.
- **Efectos laterales**: las acciones pertenecen al cliente

Venta de acciones

- **Prioridad**: media
- **Estabilidad**: media
- **Descripción**: el cliente solicita la venta de sus acciones, por lo que se lo comunica al empleado 'Broker' del SoftBank. El 'Broker' se introduce en el mercado de valores de la Bolsa de Madrid y vende las acciones del cliente.
- **Entrada**: acciones del cliente
- **Salida**: capital obtenido por las acciones
- **Origen**: petición del usuario
- **Destino**: mercado de valores (la Bolsa)
- **Necesita**: acciones
- **Acción**: venta de valores/acciones
- **Precondición**: el cliente debe de poseer acciones
- **Postcondición**: Se han vendido las acciones
- **Efectos laterales**: El capital obtenido es propio del cliente

Asesoramiento sobre acciones

- **Prioridad**: media
- **Estabilidad**: media-baja

- **Descripción:** el cliente solicita un asesoramiento sobre el mercado de valores. El empleado 'Broker' le comunica cuáles son los buenos valores con sus últimos movimientos en el mercado.
- **Entrada:** petición de asesoramiento
- **Salida:** asesoramiento
- **Origen:** petición del usuario
- **Destino:** empleado 'Broker'
- **Necesita:** información sobre el mercado de valores
- **Acción:** asesoramiento general sobre el mercado de valores (la Bolsa).
- **Precondición:** ninguna
- **Postcondición:** información transmitida del 'Broker' al cliente.
- **Efectos laterales:** creación de un texto en papel con la información transmitida.

Solicitud de hipoteca

- **Prioridad:** alta
- **Estabilidad:** media-alta
- **Descripción:** petición por parte de un cliente del banco de una hipoteca
- **Entrada:** petición del cliente del banco
- **Salida:** llamada a la siguiente función (Comprobación viabilidad de una hipoteca)
- **Origen:** petición del usuario
- **Destino:** sistema
- **Necesita:** base de datos de usuarios
- **Acción:** Solicitar hipoteca
- **Precondición:** el cliente debe cumplir una serie de requisitos y documentos, a parte de una cuenta del tipo cliente
- **Postcondición:** no tiene
- **Efectos laterales:** se llama a la siguiente función

Comprobación de la viabilidad de una hipoteca

- **Prioridad:** alta
- **Estabilidad:** media-alta

- **Descripción:** analiza el capital del cliente, su fuente de ingresos y los documentos entregados para comprobar si es posible concederle esa hipoteca.
- **Entrada:** solicitud de la hipoteca
- **Salida:** aceptación/denegación de la hipoteca
- **Origen:** petición del usuario
- **Destino:** sistema
- **Necesita:** base de datos de usuarios
- **Acción:** validación de hipoteca
- **Precondición:** el cliente debe haber hecho una petición de una hipoteca y haber entregado los documentos requeridos
- **Postcondición:** necesidad realizar pagos mensualmente
- **Efectos laterales:** se guarda la cuenta en la base de datos

Asesoramiento bancario general

- **Prioridad:** alta
- **Estabilidad:** media-alta
- **Descripción:** asesora y ofrece posibilidades financieras al usuario en función de su edad e ingresos
- **Entrada:** petición usuario
- **Salida:** opciones financieras aconsejadas
- **Origen:** petición del usuario
- **Destino:** sistema
- **Necesita:** base de datos de usuarios
- **Acción:** asesoramiento bancario
- **Precondición:** el cliente debe tener fuente de ingresos y una cuenta del tipo cliente
- **Postcondición:** no tiene
- **Efectos laterales:** no tiene

Actualización de una cartilla

- **Prioridad:** alta
- **Estabilidad:** alta
- **Descripción:** actualiza la cartilla de una cuenta en función de las últimas operaciones realizadas en ella

- **Entrada**: fichero cartilla
- **Salida**: fichero cartilla actualizado
- **Origen**: petición del usuario
- **Destino**: fichero de la cartilla del usuario
- **Necesita**: fichero cartilla
- **Acción**: actualizar cartilla
- **Precondición**: el cliente debe tener una cuenta en el banco
- **Postcondición**: debe haber realizado algún movimiento
- **Efectos laterales**: no tiene

3.3 Requisitos de rendimiento

La aplicación solo podrá ser ejecutada por un terminal. El número de usuarios registrados en ella será indefinido, mediante el uso de memoria dinámica podremos redimensionar la lista de los usuarios en función de cuál sea la cantidad de los mismos.

3.4 Requisitos sobre la persistencia de datos.

La información almacenada será un conjunto de ficheros en los que se almacenarán, de manera independiente, usuarios junto a sus datos y todas las acciones realizadas por la aplicación.

3.5 Restricciones de diseño

El diseño se hará con un lenguaje orientado a objetos en este caso Java, se debe utilizar herramientas de software libre y poderse ejecutar tanto en Windows como en GNU/Linux y Mac OS X.

El usuario podrá interactuar con el sistema.

Las interfaces deben ser intuitivas y fáciles de utilizar, de forma que tras unas breves instrucciones los usuarios sin experiencia previa puedan manejarlas sin problemas.

La aplicación deberá funcionar bajo un hardware que soporte las restricciones anteriores, es decir, no funcionará en móviles o tabletas.

3.6 Atributos del sistema software

Es una descripción sobre las características que tendrá el producto desarrollado en cuanto a:

- Fiabilidad: alta, al no depender de internet, se reduce el riesgo de fallos
- Disponibilidad: la disponibilidad ha de ser total ya que la aplicación estará en funcionamiento constantemente y no pueden producirse fallos en ningún momento
- Seguridad: al tratarse de una aplicación cerrada sin tener conexión a internet y a la que solo tienen acceso los operarios del banco, no presenta vulnerabilidades
- Mantenibilidad: cada cierto tiempo se harán revisiones técnicas formales para asegurar el buen funcionamiento del sistema
- Portabilidad: al tratarse de una aplicación en lenguaje Java sin necesidad de instalación, podrá portarse a cualquier sistema operativo que disponga de una máquina virtual Java