

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Programación Orientada a Objetos

Primer Proyecto Programado: Administrador de préstamos

Semestre II, 2015

Profesor: Mauricio Avilés

Integrantes:

Diego Tenorio Solís

Dylan Rodríguez Barboza

Resumen ejecutivo

La descripción de la solución de un problema no se debe tomar brevemente, es importante profundizar los métodos y pasos que se utilizaron para llegar a una respuesta clara y eficiente. Por esto, surge la necesidad de descripciones clave que influyan en la persona que necesite consultar su historia. En este caso, se plantean las necesidades más comunes a la hora de plantear un problema, más comúnmente dicho, las aplicaciones de fines comerciales, la cual siguió una serie de procedimientos para lograr satisfacer al que la requiera.

Podemos pensar que una creación de ese tipo no es del tanto impresionante, pero, luego de una serie de algoritmos y escritura de código, se puede profundizar su proceso de creación y, por lo tanto, despertar el interés de quienes fijaron la atención en algún proyecto o aplicación.

Es por esto por lo cual es más común la interacción directa sobre un trabajo, ya que el interés por el conocimiento abarca un aprendizaje más profundo y, por lo tanto, una puerta hacia la interacción entre personas.

Introducción

En numerosas ocasiones, cientos de personas se han planteado ideas que abarquen sus necesidades, entre dichas ideas se toma en cuenta y, como una de las principales ideas, la organización personal, ya sea de artículos que se estén vendiendo, objetos que se estén coleccionando o hasta software desorganizado.

Gran cantidad de entidades superiores han requerido habilidades que permitan construir de manera efectiva alguna forma de organizar la información de forma clara y concisa. Pero, solamente la necesidad de un servicio específico no soluciona directamente un problema que se ha planteado en la vida cotidiana.

Es por esta razón por la cual se ha intentado implementar una aplicación únicamente para este fin, en dicho caso, un fin comerciante. Dicha implementación ha intentado abarcar las necesidades básicas de un comerciante de ítems: la principal necesidad de una persona de este tipo es la exitosa organización de ítems con sus respectivas categorías y tipo, con el fin de mantener un orden, el cual es esencial para los tratos que se pueden generar con dicha aplicación.

Luego de la organización adecuada de los ítems elegidos por el usuario principal, se le debe permitir crear un préstamo de forma fácil y clara, para esto se ha incluido la capacidad para crear dicha necesidad que ha sido solicitada por los principales usuarios de este tipo de aplicación.

Ya que un principio que se debe seguir es la responsabilidad a la hora de ser acreedor de un préstamo, de acuerdo a las necesidades y requerimientos del usuario, se implementó un sistema de alarma que

avisará el día en el que un préstamo ha caducado, reforzando así uno de los principales valores como lo es la responsabilidad.

Finalmente, un adecuado manejo de datos es posible mediante una consulta de forma general a los registros hechos, por lo que el manejo de datos de forma alfabética le facilita al usuario la comprensión de información gracias a su claridad.

Presentación y análisis del problema

Son cada vez más la cantidad de personas que presentan un problema al mundo, pero, de igual forma, aumenta el número de personas que resuelven dichas inquietudes, un mínimo detalle puede ser consultado con seguridad, ya que, siempre existirá alguien que quiera enfrentar la situación y mejorarla de la manera más eficiente y cómoda para la persona que lo solicita.

Es por esta razón por la cual se ha podido disminuir gran cantidad de problemas que se presentan en la vida cotidiana, esto no quiere decir que se ha determinado el fin de las inquietudes que pueda tener un usuario, sino que se presenta el exitoso tratamiento de estos, sin un fin determinado.

Con el paso de los años han surgido más y más personas con fines comerciales que requieren de un cierto tipo de orden para manejar sus ideas con un público reservado. Es esta la principal razón por la que nace la implementación de una aplicación que, principalmente, organiza los ítems que posea el usuario de diferentes formas, ya sean categorías, tipos o personas involucradas en un préstamo de dichos ítems.

Entre los problemas que se planean solucionar, se encuentra la facilitación y comprensión clara de información para un usuario determinado que desee hacer uso de la aplicación, además de un confiable ambiente de organización tanto para el usuario principal como para los clientes que obtendrá por medio del uso del programa.

Otro problema planteado fue el manejo de valores como la responsabilidad, ya que, el préstamo de un ítem no solo incluye un

préstamo y una devolución, sino que implica más, tomando como ejemplo una devolución en el tiempo programado, es por esto que la implementación de una alarma que regule los tiempos entre préstamos es importante cuando se requiere un adecuado sentido de la responsabilidad.

Finalmente, un problema muy común que se presenta a la hora de utilizar aplicaciones de este tipo, es el manejo y comprensión de información, principalmente a la hora de consultarla, ya que, si se tiene un inadecuado manejo de esta, la consulta puede tomar horas ser comprendida y darle un uso adecuado, es por esto que actualmente se requiere un cierto patrón de orden a la hora de elaborar un programa.

Dichos problemas pueden ser resueltos mediante la programación, también el principal medio por el cual se está resolviendo el problema, específicamente la programación orientada a objetos, ya que esta requiere un orden mayor, lo que implica menos tiempo para implementar código que se encargue de mejorar o resolver un problema planteado.

La organización se maneja mediante la estructuración de datos con objetos, los cuales se encargan de guardar información específica, ya sea de los ítems, categorías, tipos o personas, los cuales serán consultados repetidas veces para la consulta, creación y edición de datos en el registro general de usuario.

Para la facilitación y almacenamiento de información, se implementó un sistema de almacenamiento de datos en el disco duro, con el fin de habilitar la posibilidad de manejo de datos a toda hora y con seguridad

de guardado que permita trabajar con la aplicación en diferentes períodos del día.

El manejo de interfaz gráfica implica una facilidad más para el usuario que haga uso de la aplicación, ya que esta se encargará de orientar al protagonista del programa hacia sus respectivos objetivos, a la vez específicos, por lo cual debe existir claridad en la interfaz de usuario, con tal de facilitar una acción.

Adicionalmente, se puede afirmar que la resolución de un problema es la puerta para el comienzo de otro, por lo que se debe poseer la habilidad y creatividad para adaptarse e innovar tecnológica y lógicamente los requerimientos actuales que cual entidad puede solicitar.

Detalladamente, se ha intentado la implementación adecuada de un sistema de creación de categorías y tipos, para una futura asignación a los ítems que el usuario quiera crear, además, el usuario puede hacer una lista de personas a las cuales les pueda hacer préstamos, a dichas personas se les podrá cambiar sus datos principales, al igual que a los ítems, tipos y categorías, creando una relación muy específica que le permita al usuario tener un orden específico para su grupo de pertenencias, evitando la pérdida de detalles que perjudiquen la interacción con sus clientes. Tanto las categorías, personas, ítems y tipos poseerán un orden alfabético cuando el usuario requiera una consulta general de estos, facilitando así la comprensión y búsqueda de información específica.

Luego de dichos procesos, y con al menos un ítem y una persona en el registro del usuario principal, se podrá realizar un préstamo, donde, si el usuario o el cliente lo requieren, podrán acordar una fecha límite para su devolución, esto se efectuará mediante una alarma que de un aviso en la fecha indicada por el usuario para un préstamo en específico. Dicha alarma puede ser recurrente, la cual hará un aviso cuando el usuario inicie la aplicación, o bien puede ser única, que se encargará de hacer el aviso en la fecha límite para entregar el ítem.

Cada parte del registro, ya sea el ítem, la categoría, el tipo o la persona, podrán ser reportadas en orden alfabético, para así facilitar la comprensión por parte del usuario de la aplicación, dichos reportes contendrán reportes básicos de la sección del registro que se seleccionó.

Ciertamente, la implementación no es completa en sí, ya que es posible la mejora de la aplicación con el paso del tiempo. En su sistema de creación de ítems, categorías, tipos y personas, podría solicitar información más específica para manejar un mejor orden de los datos, en el caso de un usuario que cumpla con ciertas habilidades de organización.

Además, la interfaz gráfica podría cambiar de acuerdo a las necesidades o gustos de un usuario, por lo que el mantenimiento de dicha sección de la aplicación es candidata para mantenimiento frecuente, al igual que los datos que se deben manejar.

De igual forma, en la sección de consulta de datos, se podría manejar los datos de acuerdo a las necesidades del cliente, ya sea de forma

alfabética, con búsquedas aproximadas de nombres o apellidos, al igual que los códigos.

Efectivamente, dichas implementaciones podrían hacerse realidad con el transcurso de los meses, ya que, de una buena idea nacen muchas más, por lo que no es imposible continuar con el proyecto en un futuro, gracias al mantenimiento que se le puede dar de forma continua y precisa.

Adicionalmente y como forma de reforzamiento, se han requerido y solicitado nuevas aplicaciones con el mismo fin, pero con diferente implementación, por lo que un trabajo en conjunto entre creadores podría tomar el inicio de diferentes y renovados programas con fines cada vez más específicos.

Por dicha razón, se ha logrado satisfacer las necesidades que surgen de los diferentes usuarios que desean manejar un programa, evidenciando que el trabajo en equipo con ideas similares mejora la interacción y aprendizaje tanto de los programadores como del usuario.

CLASE: Controladora
<p>Atributos:</p> <ul style="list-style-type: none"> -<u>Items</u> -<u>Categorias</u> -<u>Tipos</u> -<u>Personas</u> -<u>Prestamos</u>
<p>Métodos:</p> <ul style="list-style-type: none"> +crearItem (nombre:String, código:String, descripción:String, i índiceTipo:int, indCategorias: ArrayList<Integer>): void +editarItem (nombre:String, código:String, descripción:String, tipo:String, índice:int, nomCategorias:ArrayList<String>): void +eliminarItem (índice:int, nomCategs:ArrayList<String>, nombreTipo:String): void +devolverItem(índice:int): Items +devolverArrayListItems(): ArrayList<String> +crearCategoria(nombre:String): void +editarCategoria(nombre:String, índice:int): void +eliminarCategoria(índice:int, nomItems:ArrayList<String>): void +devolverCategoria(índice:int): Categorias +devolverArrayListCategoriasStr():ArrayList<String> +confirmarVacioCategorias(): boolean +agregarTipo(nombre:String): void +editarTipo(nombre:String, índice:int): void +borrarTipo(índice:int, nombrelItems:ArrayList<String>): void +devolverTipo(índice:int): Tipos

```
+devolverArrayListTipos(): ArrayList<String>
    +getTipos(): ArrayList<Tipos>
+agregarPersona(nombre:String, telefono:String, correo:String): void
    +editarPersona(nombre:String, telefono:String, correo:String,
        indice:int):void
    +eliminarPersona(indice:int): void
    +devolverArrayListPersonas():ArrayList<String>
    +devolverPersona(indice:int):ArrayList<Personas>
    +crearPrestamo(numPersona:int, ítems:ArrayList<Items>): void
+crearPrestamoConAlarma(numPersona:int, items:ArrayList<Items>,
    alarma:LocalDateTime, recurrente:boolean): void
    +getItems():ArrayList<Items>
    +devolverArrayListPrestamos():ArrayList<String>
    +devolverPrestamo(indice: int): Prestamos
+eliminarItemPrestamo(indicePrestamo:int, indiceltem:int):
    +eliminarPrestamo(indicePrestamo: int):
        +reporteCategorias(): String
        +reporteTipos(): String
        +reporteltems(): String
        +repoteUsuarios(): String
        +Escribir(): void
        +Leer(): void
```

CLASE: Alarma
Atributos: -fecha: LocalDateTime - activada: boolean -recurrente: booleann
Métodos: +Alarma(fecha: LocalDateTime, recurrente: boolean) +fueraDeTiempo:boolean +getFecha: LocalDateTime +isActivada: boolean +setActivada(activada:boolean): void

CLASE: Categorías
Atributos: -nombre: String -items: ArrayList<Items>
Métodos: +Categorías(nombre:String) +getNombre: String +setNombre(nombre:String): void +anadirItem(objeto:Items): void +devolverArrayListItems: ArrayList<Items> +eliminarItem(objeto:Items): void +getItems: ArrayList<Items>

+confirmarItemVacio: boolean
+toStringCategorias: String

CLASE: Items

Atributos:

-pres: Prestamos
-nombre: String
-codigo: String
-descripcion: String
-tipo: Tipos
-prestado: boolean

Métodos:

+Items(nombre:String, código:String, descripcion:String, tipo:Tipos)
+getCategorias(): ArrayList<Categorias>
+Itemsvacio: boolean
+getNombre: String
+setNombre(nombre:String):void
+getCodigo: void
+setCodigo(código:String):void
+getDescripcion: String
+setDescripcion(descripcion:String): void
+getTipo: Tipos
+setTipo(tipo:Tipos): void
+eliminarCategorias:void
+eliminarCategoria(categ:Categorias): void

```

+devolverArrayListCategorias: ArrayList<String>
+eliminarCategoria(categ:ArrayList<Categorias>):void
    +devolverNombreTipo: String
        +eliminarTipo: void
        +getPrestado: boolean
        +setPrestado(prestado): void
        +toStringTipo:String

```

CLASE: Personas

Atributos:

```

-nombre: String
-telefono: String
-correo: String
-prestamoPendiente: boolean
-prestamos: ArrayList<Prestamos>

```

Métodos:

```

+Personas(nombre:String, teléfono:String, correo:String)
+añadirPrestamo(préstamo:Prestamos): void
+Personasvacio: boolean
+getNombre: String
+setNombre(nombre:String):void
+getTelefono: String
+setTelefono(teléfono:String):void
+getCorreo:String
+setCorreo(correo:String):void
+getNumPrestamos:int

```

```
+getPrestamos:ArrayList<Prestamos>
+eliminarItem(préstamo:int, ítem:int):void
+eliminarPrestamo(préstamo:int):void
```

CLASE: Prestamos

Atributos:

```
-items: ArrayList<Items>
-alarma: Alarma
-persona: Personas
```

Métodos:

```
+preparado(ítems:Items, personas:Personas):boolean
+Prestamos(ítems:ArrayList<Items>)
+Prestamos(ítems:ArrayList<Items>, alarma:LocalDateTime,
recurrente:boolean)
+devolverArrayListItems:ArrayList<String>
+eliminarItem(índice:int): void
+eliminarItems: void
+toStringItems: String
+toStringAlarmas: String
```

CLASE: Tipos

Atributos:

-nombre:String

-items: ArrayList<Items>

Métodos:

+devolverArrayListItems: ArrayList<String>

+anadirItem(objeto:Items):void

+eliminarItem(obj:Items):void

+getNombre:String

+setNombre(nombre:String):void

+Tipos(nombre:String)

+getItems:ArrayList<Items>

+setitems(items:ArrayList<Items>):void

+toStringTipos:String

+getItemsOrdenados:ArrayList<Items>

Conclusiones

El trabajo en equipo mejora la comprensión mutua sobre los requerimientos de un problema, lo que facilita la búsqueda de una solución mutua que resuelva el problema presentado.

La plataforma NetBeans facilita la resolución del problema, en los ámbitos tanto lógicos como gráficos, por lo que se facilita el manejo de herramientas gráficas que se adaptan fácilmente a la lógica de un programa.

El manejo de código mediante el paradigma de programación orientada a objetos facilita el manejo de datos mediante clases y objetos, ya que gracias a los atributos que posee una determinada clase es posible el guardado organizado de información.

El problema que presenta un usuario no será exclusivamente de él, la solución no implica a un solo beneficiado, sino numerosas personas que se plantearon el mismo problema.

La finalización de un programa no significa que este no pueda ser mejorado después, por lo que un mantenimiento adecuado puede solucionar nuevos problemas y mantener al margen los que dieron origen a la solución principal.

Las pruebas durante la implementación de soluciones son esenciales a la hora de resolver un problema, ya que, con su ausencia, será fácil fallar hasta la tarea más simple, se deben realizar repetidas pruebas para hallar errores más fácilmente.

Recomendaciones

Organización adecuada a la hora de manejar la presión de un trabajo en equipo, de esta forma se fortalecerán los lazos, mejorando la productividad del equipo.

Utilización de herramientas que faciliten el trabajo, tanto en la parte lógica como con la interfaz, por ejemplo: la plataforma NetBeans incluye interfaz de java swing que facilita el manejo de la interfaz de usuario.

Manejo del paradigma de programación orientado a objetos, ya que este facilita la organización de la información mediante la implementación de las clases y objeto que se quieran definir.

La solución de un problema de forma individual es aceptable, pero el procesamiento y resolución de problemas en conjunto hará una relación específica que será capaz de solucionar las inquietudes que se han presentado a lo largo de la elaboración de un bien común.

Una vez finalizado un trabajo o proyecto, ya sea individual o en equipos, se debe tener documentación de lo hecho, ya que, es muy probable de que haya necesidad de dar mantenimiento, por lo que el reciclaje de código en este caso estimula el ahorro de tiempo y la efectiva participación de los miembros de un subgrupo.

Una vez seguros de una funcionalidad específica, es recomendable y probable la realización de pruebas en pequeños fragmentos que garanticen el funcionamiento adecuado de un programa, que deberá ser presentado en limpio para la persona que lo solicitaba.

Referencias

Serialización en Java. Puntocomnoesunlenguaje. Fecha de consulta: 13:30, Septiembre 16.

Desde:

<http://puntocomnoesunlenguaje.blogspot.com/2013/10/java-serializacion-persistencia.html>