

Tecnológico de Costa Rica

Manual Técnico de la Aplicación Móvil

Simplex Educativo

Profesora: María Estrada.

Curso: Proyecto de Ingeniería de Software.

20.01.2017

Jose Fernando Molina Chacón

Yordan Jiménez Hernández

TABLA DE CONTENIDOS

Introducción	3
Propósito	4
Alcance	5
Definiciones, acrónimos y abreviaturas	5
Referencias	6
Perspectiva del sistema	6
Características del sistema	6
Arquitectura del Sistema	9
Diseño del sistema	10
Metodología del Diseño	10
Principios SOLID	10
Implementación de MVC.	11
Herramientas gratuitas.	12
Exclusión de la eficiencia.	12
Portabilidad	12
Estándar de Programación	12
Herramientad de Desarrollo de Software	12
Android Studio 2.1	12
Draw.io	13
Descomposición del Sistema	14
Paquetes Arquitecturales	14
Modelo	14
Modelo.parser	15
Java_cup	15
Dto	15
Controlador	15
Vista	16
Clases	16
Descripción de Componentes	17
dto.Point	17

Tipo	17
Propósito	17
Función	17
Restricciones	17
Dependencias	17
Interfaces	17
Recursos	17
Procesamiento	18
Datos	18

Introducción

Durante el transcurso del curso de Investigación de Operaciones de la carrera de Ingeniería en Computación impartida en el Instituto Tecnológico de Costa Rica, se abarcan temas de maximización y minimización de costos, métodos matemáticos que pueden ser computarizados para su utilización en diversos campos. En la educación, un método utilizado en el curso nombrado anteriormente es el método Simplex. Dicho método es ampliamente conocido, y existen herramientas de software como Lindo o páginas web PhpSimplex que proveen un programa para solucionar dichos problemas.

Sin embargo, estos programas poseen grandes limitaciones a la hora de utilizarlos como herramienta educativa. Algunos de ellos solamente solucionan el problema, sin dar información que puede ser relevante para el profesor o estudiantes. Por otra parte, cada software representa los problemas de programación lineal por medio de una matriz con un formato diferente, por lo que se vuelve difícil interpretar los resultados o datos provistos por estas herramientas. Es por estos motivos que para la enseñanza en el campus estas herramientas no poseen las características que faciliten el aprendizaje de los estudiantes o ayuden al transcurso de la clase impartida por el profesor encargado del curso.

El programa Simplex Educativo se centra en la implementación de una solución computacional que solucione problemas de programación lineal por medio del algoritmo simplex de una manera dinámica y amigable para el aprendizaje, donde el usuario puede indicar distintas formas de solucionar un ejemplo de maximización o minimización, con el fin que el usuario pueda resaltar características relevantes a los estudiantes acerca de comportamientos que tome el algoritmo simplex ante los diferentes escenarios posibles.

El usuario podrá ingresar textualmente un problema de programación lineal con un formato definido, y el programa podrá identificarlo y resolverlo de manera que el usuario pueda entender cada uno de los pasos que implica la resolución de un problema de este tipo. Debido a la naturaleza educativa del producto, no hay restricciones en cuanto a eficiencia del programa. La solución debe ser portable a través de diferentes sistemas operativos, por lo tanto será desarrollada en el lenguaje de programación Java, debido a que este lenguaje no es dependiente de la plataforma en que se ejecute. El sistema no poseerá manejo de usuarios ni autenticaciones.

Se desarrollaron conjuntamente dos sistemas: Una aplicación para escritorio y una aplicación móvil para dispositivos Android. Ambos sistemas poseen la funcionalidad básica de interpretar un problema ingresada a manera de texto y resolverlo utilizando el método simplex. Sin embargo, la aplicación de escritorio poseerá más funcionalidades que la versión móvil. La versión móvil solamente poseerá la capacidad de mostrar los pasos intermedios a modo de tabla resumen, o

llegar de manera directa a una solución. Por su parte, la versión de escritorio podrá resolver un problema de programación lineal entero (esto es, donde su solución sean números enteros) mediante el algoritmo de Gomory o Branch and Bound, ingresar una matriz numérica textual en lugar de un problema, escoger la posición en la cual se va a pivotar en cada paso del algoritmo, cambiar entradas de la matriz una vez iniciado el algoritmo y agregar restricciones, filas o columnas a la matriz de cada paso.

Propósito

El propósito de este documento es proveer a las partes interesadas en el funcionamiento del sistema Simplex Educativo con la mayor cantidad de información posible acerca de la de la manera en que fue implementada la solución. Se tratará de describir en detalle todos los componentes de software involucrados, la arquitectura y las decisiones que se tomaron durante el desarrollo de producto, la interacción entre las diferentes partes del software y en general todo el conocimiento que tiene el equipo de desarrollo que pueda ser utilizado por personas ajenas al equipo para modificar o extender la solución. Se espera que este documento pueda ser una referencia para desarrolladores que quieran comprender a fondo la implementación del software Simplex Educativo, además se espera que las personas que hayan leído el documento sean capaces de comprender más sencillamente el código fuente del programa, y de esta manera poder realizar futuros cambios, extensiones o cualquier tipo de modificación al producto de una manera sencilla, disminuyendo así el tiempo de aprendizaje acerca de la implementación de un nuevo producto de software.

Alcance

En este documento se posee el objetivo de explicar la aplicación móvil para el sistema operativo Android que ayude al profesor PhD. José Elías Helo a desarrollar su clase de Investigación de Operaciones de una manera dinámica, que propicie el entendimiento de la materia por parte del estudiante. Una vez desarrollada, esta aplicación será utilizada por el profesor solamente y será propiedad total y exclusiva de él, incluido el código fuente del programa.

El proyecto incluye documentación técnica en forma de comentarios en el código fuente, manual de usuario y documentación externa. También incluye una serie de artefactos de aseguramiento de la calidad como pruebas unitarias y de integración, así como pruebas de sistema y pruebas de aceptación que serán realizadas por el cliente.

El proyecto no incluye el mantenimiento de la aplicación una vez entregado el producto. Cuando se haga la entrega final del proyecto, el cliente se hará

responsable del producto de ese momento en adelante y el equipo de desarrollo ya no será responsable del mismo.

La construcción del proyecto está estimada en un plazo no mayor a 35 días. La aplicación móvil podrá ser ejecutada en dispositivos con el sistema operativo Android en su versión 6.0 o superior.

Definiciones, acrónimos y abreviaturas

- **ERS:** Especificación de Requerimientos de Software.
- **MVC:** Model-View-Controller. Patrón de diseño arquitectural.
- **Método Simplex:** Algoritmo creado por George Dantzig para resolver problemas de programación lineal.
- **Programación Lineal:** Modelo matemático para buscar la optimización de una función objetivo que cumpla con N restricciones al mismo tiempo.
- **Java:** Lenguaje de programación orientado a objetos.
- **JDK:** Java Development Kit. Herramientas de desarrollo del lenguaje Java.
- **UML:** Unified Modeling Language. Lenguaje gráfico descriptivo para uso en procesos de software.
- **Problema Infactible:** Problema de programación lineal que debido a sus restricciones hace que la función objetivo no se pueda cumplir bajo ninguna circunstancia.
- **Problema No Acotado:** Problema de programación lineal que debido a sus restricciones hace que la función objetivo pueda ser maximizada o minimizada de manera infinita (no existe límite para detenerse).
- **BVS:** Basic Variables. Variables básicas, es decir, variables que poseen un valor definido en cierto momento.
- **RHS:** Right Hand Side. Lado derecho de una igualdad o desigualdad. En representación matricial simplex, la última columna de derecha a izquierda.
- **Radios:** Resultado de dividir el RHS entre una columna N de una matriz simplex. Tienen la particularidad de que si la división es entre 0 o el resultado es un número negativo, el resultado de los radios será infinito.
- **Pivatar:** Tomar una entrada diferente de cero de una matriz y realizar operaciones fila de manera que la columna de esa entrada sea básica, es decir, solamente posea 0's y solamente un 1 en la posición de pivote escogida.
- **Android:** Sistema operativo móvil desarrollado por Google.
- **Scanner:** Programa computacional que tiene por entrada una cadena de caracteres. Toma esta cadena de caracteres y la divide en "tokens", estos son una unidad de texto que tiene algún valor o significado implícito.
- **Parser:** Programa computacional que tiene por entrada una cadena de caracteres. Trabaja en conjunto con el scanner para validar que la cadena

cumpla un formato específico y extrae la información relevante de dicho formato.

- **LALR**: Look-Ahead Left to Right. Un tipo de parser.

Referencias

1. Documento de Visión de Simplex Educativo.
2. Especificación de Requerimientos de Software de Simplex Educativo.
3. Documento de Arquitectura de Software Simplex Educativo.
4. Plan de Pruebas de Simplex Educativo.
5. Manual de Usuario de Simplex Educativo.
6. Manual de Usuario de JFlex: <http://jflex.de/manual.html>.
7. Manual de Usuario de Java CUP: <http://www2.cs.tum.edu/projekte/cup/manual.html>

Perspectiva del sistema

Características del sistema

En el siguiente apartado se describirán las características del sistema Simplex Educativo en su versión de aplicación móvil. Estas características fueron tomadas del documento “Especificación de Requerimientos de Software de Simplex Educativo”.

Obtener de manera inmediata una solución óptima de un problema de programación lineal.

Descripción

Los problemas de programación lineal buscan maximizar o minimizar una función objetivo con N variables lineales sujetas a M inecuaciones lineales que incluyen las variables de la función objetivo. El método Simplex es un algoritmo desarrollado por George Dantzig para resolver esta clase de problemas. Ellos se utilizan en diversos campos de la matemática aplicada e ingenierías para encontrar soluciones a problemas que puedan ser modelados mediante el método de programación lineal. La aplicación móvil será capaz de solucionar un problema de programación lineal con N variables y M

restricciones lineales por medio del método Simplex, brindando inmediatamente la solución al usuario.

Prioridad

Alta

Secuencia Estímulo/Respuesta

1. La aplicación muestra el apartado donde se ingresa el problema de programación lineal.
2. El usuario ingresa el problema de programación lineal por medio de una cadena de texto.
3. El usuario escoge el formato numérico en el cual desea la respuesta.
4. El usuario selecciona la opción de solución inmediata.
5. La aplicación interpreta el problema ingresado y obtiene la primera solución óptima que genere el método Simplex.
6. La aplicación muestra el resultado óptimo final del problema.

Requerimientos Funcionales

MREQ-1 Interpretar un problema ingresado:

El sistema debe interpretar dentro del problema de programación lineal ingresado por medio de una cadena de texto por parte del usuario los valores enteros, valores decimales o fraccionarios, variables y restricciones que se encuentren representados en dicha cadena. En caso de que no contenga el formato esperado, el sistema le indicará al usuario por medio de un mensaje la naturaleza del error.

MREQ-2 Solución por medio del Simplex:

El sistema debe utilizar el método Simplex común desarrollado por George Dantzig y enseñado por el profesor José Helo en su clase de investigación de operaciones para obtener la primera solución óptima de un problema de programación lineal.

MREQ-3 Resultados fraccionarios o decimales:

Los valores numéricos brindados por la ejecución de la aplicación deben ser mostrados en formato decimal o fraccionario. El formato decimal será representado mediante 2 decimales, mientras que el formato fraccionario será representado mediante una fracción con numerador y denominador enteros. La elección del formato será por parte del usuario antes de ejecutar la solución del problema.

Listar los pasos intermedios para encontrar la solución a un problema de programación lineal.

Descripción

El algoritmo simplex representa matricialmente (esto es, por medio de una matriz numérica) el problema de programación lineal de modo que cada columna representa una variable y cada fila representa una restricción. Por medio de operaciones de pivote (esto significa escoger una entrada de la matriz, convertir en 1 dicha entrada y realizar operaciones fila para reducir las otras entradas de la columna a 0), el algoritmo se traslada entre las posibles soluciones hasta encontrar una solución óptima. En cada paso del algoritmo, hay una operación de pivote que cambia el estado de la matriz que representa el problema lineal.

La aplicación debe ser capaz de solucionar un problema de programación lineal por medio del método Simplex donde se pueda generar las tablas o matrices intermedias que se produjeron para obtener la primera solución óptima del problema ingresado. Esto significa que debe ser posible observar todas las tablas intermedias que fueron iteradas en cada paso para llegar a la solución final.

Prioridad

Alta.

Secuencia Estímulo/Respuesta

1. La aplicación muestra el apartado donde se ingresa el problema de programación lineal.
2. El usuario ingresa el problema de programación lineal por medio de una cadena de texto.
3. El usuario escoge el formato numérico en el cual desea la respuesta.
4. El usuario selecciona la opción de solución por pasos.
5. La aplicación muestra uno a uno cada iteración necesaria para obtener la solución óptima. El usuario avanza manualmente mediante un botón entre las iteraciones.
6. La aplicación indica y muestra que se ha obtenido la solución final óptima.

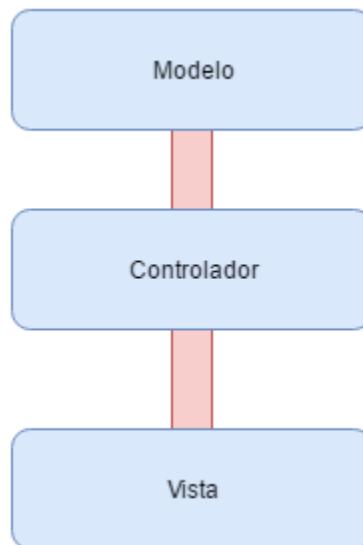
Requerimientos Funcionales

MREQ-4 Iteraciones intermedias de la solución de un problema de programación lineal

La aplicación será capaz de interpretar las iteraciones intermedias que se generaron a la hora de obtener una solución óptima de un problema de programación lineal por medio del algoritmo Simplex común y las mostrará al usuario en pantalla en forma de matriz textual. En el transcurso de dicho proceso se indicarán las características de la iteración, esto significa mostrar las variables de holgura o artificiales agregadas y las variables originales, mostrando en pantalla la representación textual de la matriz y junto con las operaciones fila que serán realizadas en cada paso.

Arquitectura del Sistema

La arquitectura del sistema está basado en la conocida arquitectura de N-Capas. Debido a que el programa por desarrollar no es significativamente complicado en términos de responsabilidades, cantidad de clases ni extensión se decidió realizar la división de las capas de acuerdo a la función que cumplen respecto al patrón Model-View-Controller. Esta estrategia ha sido escogida debido a que aisla las responsabilidades de cada clase dependiendo solamente de su función dentro del sistema respecto al patrón mencionado. La división lógica de las capas del sistema Simplex Educativo es la siguiente. El diagrama de clases completo puede ser encontrado en el repositorio.



- **Modelo:** Es la capa encargada de representar y manejar el comportamiento de los datos solamente. Ella responde a estímulos por parte de la capa controlador únicamente. Es en esta capa donde se encuentran todas las clases necesarias para la ejecución de los casos de uso a nivel lógico meramente, sin control por parte del usuario ni interfaz gráfica de por medio.

- **Controlador:** Capa encargada de controlar la interacción entre la vista y el modelo. Es esta capa la cual recibe los estímulos por parte del usuario y comunica al modelo la acción pedida, para luego tomar la respuesta del modelo y presentárselas a la vista para su despliegue.
- **Vista:** Capa encargada de la representación gráfica del modelo. Utiliza al controlador para hacer pedidos al modelo, y el controlador se encarga de actualizar la vista acordemente.

Diseño del sistema

Metodología del Diseño

El diseño del sistema Simplex Educativo fue realizado utilizando el paradigma de orientación a objetos. El diseño del programa fue realizado tomando en cuenta las siguientes restricciones:

Principios SOLID

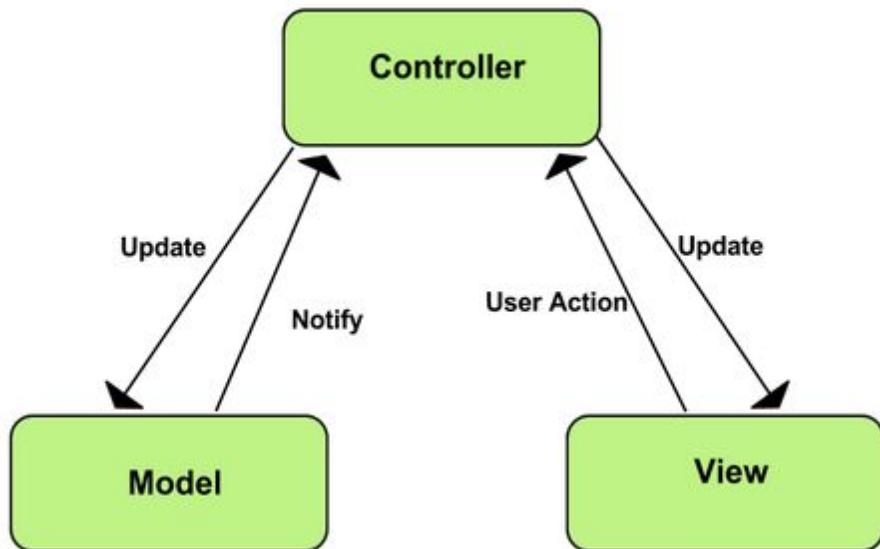
Para el diseño arquitectural se debe tomar en cuenta los principios SOLID para diseño de software. Dichos principios se describen a continuación:

- **Single Responsibility Principle:** Toda entidad de software tendrá una y solamente una responsabilidad.
- **Open-Closed principle:** Las entidades deben estar cerradas para modificación pero abiertas para extensión.
- **Liskov Substitution Principle:** Los objetos de tipo T pueden ser sustituidos por objetos de tipo S siempre que S sea un subtipo de T. Esto implica que las interfaces públicas de subclases deben ser iguales a las de su superclase.
- **Interface Segregation Principle:** Los objetos no deben depender de métodos que no utilicen.
- **Dependency Inversion Principle:** Módulos de alto nivel no deben depender de módulos de bajo nivel. Ambos deben depender de abstracciones. Asimismo, las abstracciones no deben depender de detalles. Los detalles deben depender de abstracciones. Esto implica que las clases concretas no deben tener referencias a objetos concretos, sino a sus abstracciones.

Implementación de MVC.

Para la implementación de la solución se debe utilizar el patrón arquitectural Model-View-Controller. En este patrón existen diferentes tipos de objetos que pertenecen a alguna de las categorías anteriores:

en el modelo está la lógica de negocio, en la vista las interfaces gráficas para interacción con el usuario y el controlador es el encargado de recoger las acciones del usuario en la interfaz gráfica y modificar el modelo de manera acorde. La variación del mismo será la siguiente:



Este tipo de MVC provee una mayor independencia entre el modelo y la vista, pues el modelo no debe conocer nada de la vista, de ello se encarga el controlador, el cual la actualizará acordemente dependiendo de las acciones realizadas por el usuario.

Herramientas gratuitas.

El proyecto será realizado sin remuneración económica, por este motivo todas las herramientas que se utilicen para el desarrollo del mismo deben ser gratuitas para evitar incurrir en costos innecesarios por parte del equipo de desarrollo.

Exclusión de la eficiencia.

La eficiencia del algoritmo Simplex no es una restricción. El software será utilizado para la educación y no para la resolución de problemas complicados de programación lineal, por lo cual las restricciones de eficiencia son despreciables.

Portabilidad

La aplicación móvil fue desarrollada para dispositivos que ejecuten el sistema operativo android con una versión superior a 4.0. Para su construcción se utilizó la herramienta Android Studio donde se

encuentran las librerías necesarias para compilar y ejecutar la solución brindada.

La totalidad de la estructura utilizada dentro de esta aplicación fue reutilizado de los componentes construidos dentro de la aplicación de escritorio con una pequeña variante en la estructura punto. Por ello en la mayoría de las descripciones se hace referencia al documento manual técnico.

Estándar de Programación

Para el desarrollo de la solución se utiliza el estándar de desarrollo de código de Google. Dicho estándar puede ser accedido en el siguiente hipervínculo: <https://google.github.io/styleguide/javaguide.html>

Herramientas de Desarrollo de Software

Android Studio 2.1

Para el desarrollo del proyecto se utilizó el entorno de desarrollo que propicia herramientas para aplicaciones móviles Android Studio, el cual provee una gran cantidad de herramientas gráficas y funcionales para el desarrollo de proyectos en el lenguaje de programación Java y la interacción con la parte móvil. La aplicación brinda un entorno para desarrollar la capa gráfica de la aplicación móvil a brindar por medio del SDK.

Descarga: <https://developer.android.com/studio/index.html>

Draw.io

Herramienta libre para realizar diagramas para proyectos de desarrollo de software. Incluye vasta cantidad de íconos, líneas y componentes necesarios para realizar variedad de diagramas UML útiles para la descripción de un proyecto de software. Posee la funcionalidad de trabajo colaborativo en tiempo real para múltiples usuarios.

Acceso: <https://www.draw.io/>

JFlex

Para la realización del scanner que divida la cadena de texto en los diferentes tipos de tokens reconocibles se utilizó la herramienta JFlex. Esta herramienta permite definir una serie de expresiones regulares y generar código que deba ser ejecutado cuando el scanner encuentre dicha expresión regular en el texto que está analizando. Su función principal es generar el scanner que divide la cadena de

entrada de texto en tokens que puedan ser utilizados por el parser para realizar su trabajo.

Documentación: <http://jflex.de/manual.html>

Java CUP

Para la realización del parser se utilizó la herramienta CUP. Esta herramienta trabaja en conjunto con JFlex para lograr un análisis léxico y sintáctico de una cadena de texto. Cup significa “Construction of Useful Parsers”, y es un generador de un parser LALR. El programador define los diferentes símbolos de la gramática, así como las diferentes producciones de un lenguaje libre de contexto, y este programa da la posibilidad de ejecutar cierto código cuando alguna producción es reducida, pudiendo controlar las acciones que va realizando el parser conforme se avanza por la gramática.

Documentación: <http://www2.cs.tum.edu/projects/cup/docs.php>

JUnit

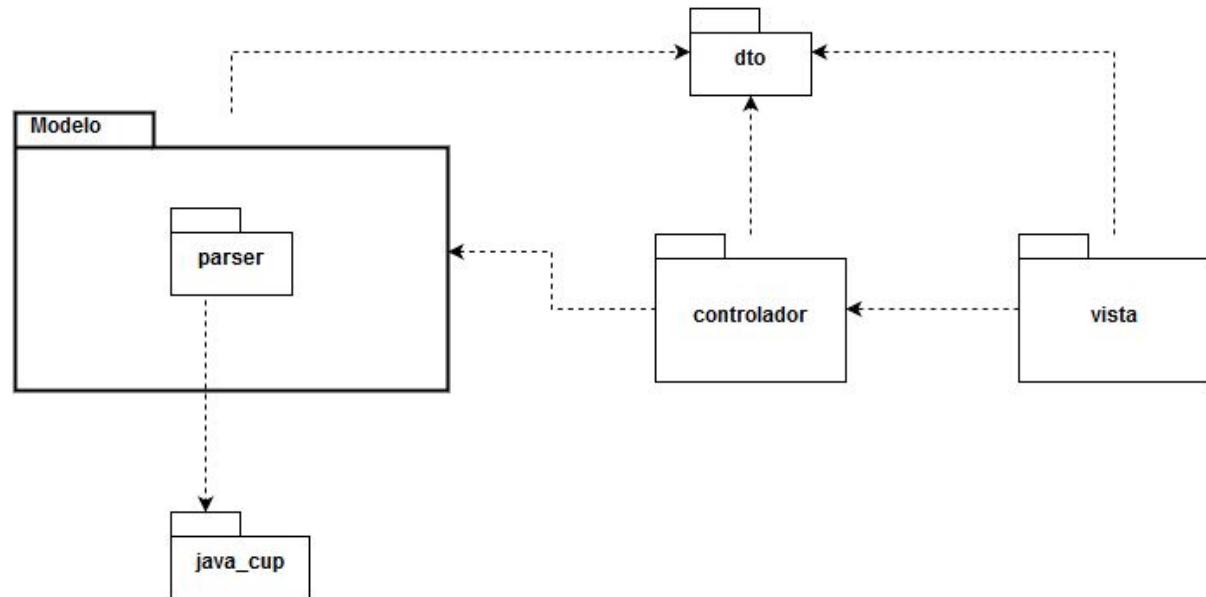
JUnit es una biblioteca utilizadas para hacer pruebas unitarias de aplicaciones Java. JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Esta herramienta fue utilizada para realizar las pruebas unitarias y de integración de la aplicación.

Página oficial: <http://junit.org/junit4/>

Descomposición del Sistema

En el siguiente apartado se busca descomponer el sistema Simplex Educativo en sus subpartes para un resumen general acerca del funcionamiento y la interacción de cada componente.

Paquetes Arquitecturales



Modelo

Contiene todas las clases que representan el modelo de la solución. Es el paquete encargado del funcionamiento lógico de la aplicación, siendo él el responsable de proveer las funcionalidades necesarias para resolver los casos de uso descritos anteriormente, siendo como objeto principal solucionar problemas de programación lineal por medio del método Simplex.

El modelo es agnóstico de su controlador y la vista. Depende del paquete dto donde se almacena un Data Transfer Object, el cual es utilizado para transferir datos entre las diferentes capas arquitecturales y para el procesamiento del problema.

Modelo.parser

Almacena dentro de su contenido los componentes utilizados para comprobar y validar la estructura de la cadena de texto ingresado por parte del usuario. La importancia de la existencia de las clases dentro del paquete rige en evitar comportamientos erróneos en la solución de un problema de programación lineal. La mayoría de las clases contenidas son autogeneradas por las herramientas Cup y JFlex que facilitan la acción objetivo del paquete, validar el formato de entrada del problema ingresado por el usuario.

Java_cup

Librería auxiliar utilizada por la clase SimplexParser dentro del paquete Modelo.parser. Java_cup brinda comportamiento de un parser LALR a la clase utilizada en la solución para validar las cadenas de texto.

Dto

Almacena objetos de transferencia de datos utilizados en los distintos paquetes de la solución creada, solo contiene una clase llamada DtoSimplex, la cual contiene toda información del problema y su representación lógica. Dicho paquete también representa al patrón de diseño Dto, con el cual se asegura que los cambios en los parámetros de los métodos de las clases causen la menor cantidad de modificaciones en cascada. Esto es de especial importancia para el mantenimiento del programa y simplifica el desarrollo, debido a que se unifica todo el estado del problema dentro de un solo objeto. Además baja los índices de rigidez y fragilidad dentro del software.

Controlador

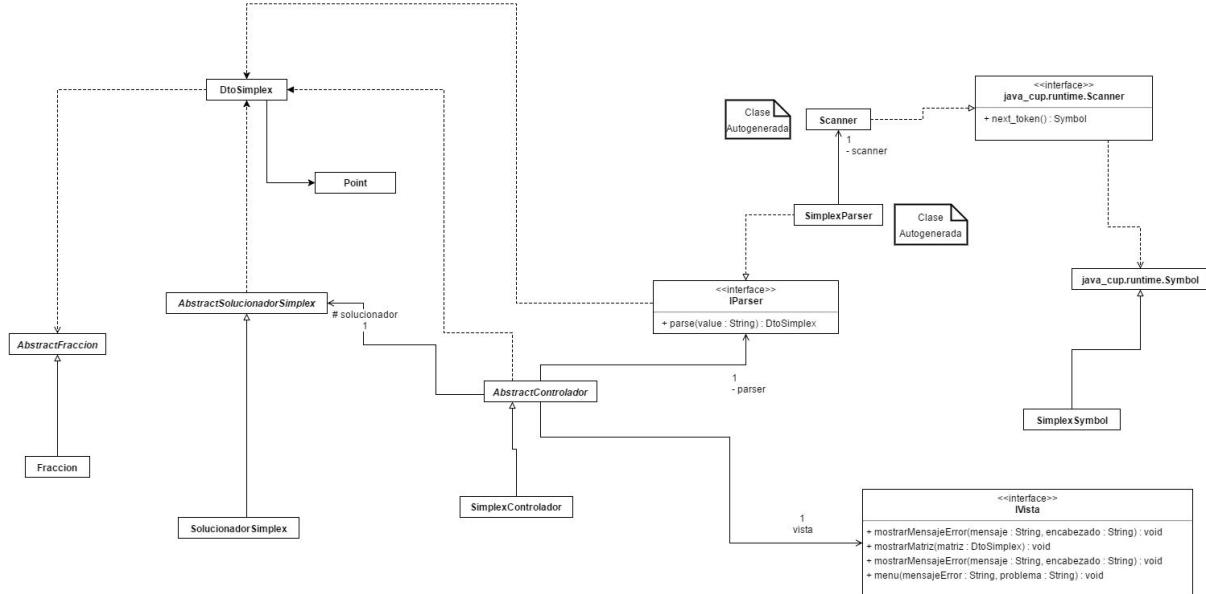
Componente esencial del patrón de diseño MVC anteriormente descrito. Como su nombre lo indica, contiene la lógica de control y ejecución del programa. El controlador es el encargado de consumir los recursos provistos por el modelo. En él se encuentran las clases encargadas de llamar a las funciones provistas por el modelo y actualizar las clases que proveen la interfaz gráfica de la solución. Provee una interfaz que puede ser consumida por el paquete vista para cumplir con los casos de uso descritos por el cliente.

Este paquete depende del otro paquete modelo solamente.

Vista

Contiene las clases que se encargan de cumplir la funcionalidad gráfica que implementará los casos de uso solicitados por el cliente. Contiene la lógica de despliegue de la información contenida en los objeto DTO (por ello la dependencia hacia el paquete DTO), y consume la interfaz provista por el paquete Controlador para cumplir su funcionalidad principal. En este paquete se encuentran las clases que crean la interfaz gráfica de la solución y la lógica de interacción con el usuario, peticiones, información o estado del problema ingresado.

Clases



A continuación se presenta la descripción de cada clase perteneciente al diagrama anterior, donde el lector podrá encontrar más información sobre cada clase y comprender el motivo de la inclusión de la clase dentro de la solución brindada. Se espera que el lector de la siguiente sección posea conocimientos del lenguaje de modelado unificado para poder comprender las relaciones entre clases, conceptos y estructuras presentes en el diagrama. En caso de que el lector no posea conocimientos en el mismo o desee informarse mejor puede consultar el siguiente enlace: <http://www.uml.org/what-is-uml.htm>

- **Point:**

- **Descripción:** Definición de un contenedor para almacenar la coordenada con una ubicación en x y y. Dichos valores enteros representan dentro del programa el valor de la columna x y el valor de la fila y, dentro de la matriz que se almacena en el **DtoSimplex**.
- **Paquete:** dto.

Las demás clases se encuentran explicadas dentro del documento denominado manual técnico de la aplicación de escritorio.

Descripción de Componentes

1. dto.Point

1.1. Tipo

Archivo .java.

1.2. Propósito

Almacenar la coordenada de pivoteo durante la resolución de un problema de programación lineal. Es utilizado durante la resolución de todos los requerimientos utilizados.

1.3. Función

Estructura que indica la ubicación en el eje “x” y “y”, para determinar dentro de la matriz donde realizar una operación de fila.

1.4. Restricciones

- Los atributos enteros “x” y “y” no pueden ser nulos.

1.5. Dependencias

No posee dependencias a ninguna otra estructura.

1.6. Interfaces

- **clone():** crea una nueva instancia de tipo de dato Point, donde se posee los mismos valores de los atributos de la instancia original.

1.7. Recursos

Esta clase no utiliza algun recurso.

1.8. Procesamiento

No posee algún procesamiento más que el establecer y retornar datos.

1.9. Datos

- **x:** valor entero que representa al eje x y en caso de la matriz de coeficientes las columnas.
- **y:** valor entero que representa al eje y y en caso de la matriz de coeficientes las filas.

Las demás componentes se encuentran explicadas dentro del documento denominado manual técnico de la aplicación de escritorio.