

Proyectos de Programación 2021/2022

Hoja de cálculo sencilla

**Cristina Bretons, Omar Tanveer,
Priyanka Amarnani, Mark Smithson**

Resumen

Se trata de construir un entorno para la manipulación de datos organizados en forma de tablas, donde se puedan efectuar operaciones con funciones. El entorno permite el tratamiento con datos de tipo numérico, textual y de tipo fecha. Además de funcionalidades como generar gráficas, calcular estadísticos, operaciones aritméticas entre bloques, etc.

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya
Barcelona, Catalunya
Abril 2022

Índice

1. Documentación del los casos de uso	2
1.1. Descripción de los casos de uso	2
1.1.1. Funcionalidad de gestión de Documento	2
1.1.2. Funcionalidad de gestión de Hoja	7
1.1.3. Operaciones	11
1.1.4. Opcionales	16
1.1.5. Referencias	19
2. Modelo conceptual de los datos	20
2.1. Descripción de las clases	20
2.1.1. Clase Documento	20
2.1.2. Clase Hoja	20
2.1.3. Clase Bloque	20
2.1.4. Clase Celda	20
3. Estructuras de datos	21
3.1. Documento	21
3.2. Hoja	21
3.3. Celda	21
3.4. Bloque	21
4. Librerías externas utilizadas	22
4.1. Digidemic UnitOf	22
4.2. Knowm XChart	22
5. Relación de clases implementadas	23

1. Documentación del los casos de uso

En esta sección se documenta de manera detallada y completa los diferentes casos de uso. Se describen las diferentes operaciones realizables con su comportamiento y los posibles errores que se pueden generar.

1.1. Descripción de los casos de uso

1.1.1. Funcionalidad de gestión de Documento

- **Nombre:** Ayuda
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica que quiere ayuda.
 2. El sistema le muestra al usuario una pequeña guía para orientarlo.
- **Posibles errores:**

- **Nombre:** Salir
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica que quiere salir de la aplicación.
 2. El sistema pregunta al usuario si quiere guardar los datos.
 3. Si no se quieren guardar se salta al **punto 7**.
 4. El sistema presenta el sistema de directorios.
 5. El sistema pregunta al usuario que nombre quiere que tenga el archivo.
 6. El sistema guarda el archivo con el nombre indicado y con extensión .pomc.
 7. El sistema cierra la aplicación
- **Posibles errores:**

- **Nombre:** Importar archivo
- **Actor:** Usuario
- **Comportamiento:**

1. El usuario indica que quiere importar un archivo.
2. El sistema presenta el sistema de directorios al usuario.
3. El usuario elige el archivo a importar.
4. Dependiendo de la extensión del archivo el sistema actuará de una manera o de otra.
 - 4.1. Extensión .txt o .csv. En este caso los archivos deberán tener las columnas separadas por puntos y comas o por solo comas. Los números con decimales tendrán que marcar su parte decimal con un '.'. Las fechas tendrán el siguiente formato: DD/MM/YYYY.
5. El sistema indica que se ha importado el archivo correctamente.

■ **Posibles errores:**

1. Si el tipo de archivo no es compatible con la aplicación, saldrá un mensaje de error y no se abrirá el archivo. Posteriormente se volverá al punto 2.

■ **Nombre:** Cargar archivo

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere cargar un archivo
2. El sistema presenta el sistema de directorios del sistema.
3. El usuario elige el archivo a cargar.
4. La extensión del archivo a cargar deberá ser .pomc
5. El sistema indica que se ha cargado correctamente el archivo

■ **Posibles errores:**

1. Si el archivo no tiene una extensión .pomc, no se cargará el archivo seleccionado. Posteriormente se volverá al punto 2.

■ **Nombre:** Guardar archivo

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere guardar el archivo que está usando en ese momento. (se guardará con la extensión .pomc)

2. El sistema sobrescribe el archivo si existe en el directorio donde se está trabajando. Posteriormente se irá al **punto 5**
3. En caso de que no exista, el sistema informará al usuario para que indique un nombre para el archivo y para que se guarde.
4. El usuario indicará el directorio donde quiere guardar el archivo.
5. El sistema indica que el archivo se ha guardado correctamente.

■ **Posibles errores:**

1. Si no hay ningún documento que guardar, se mostrará un error.

■ **Nombre:** Guardar archivo como

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere guardar el archivo que está usando en ese momento.
2. El sistema presenta el sistema de directorios del sistema.
3. El usuario indica el directorio en el cual quiere guardar el archivo.
4. El sistema le pide al usuario que le ponga nombre al documento.
5. El usuario indica el nombre del archivo
6. El sistema guarda el archivo correctamente (con extensión .pomc) y muestra un mensaje indicandolo.

■ **Posibles errores:**

1. Si el archivo tiene el mismo nombre que otro previamente guardado, el sistema le preguntará si quiere sobrescribirlo o guardarlo con otro nombre.
2. Si no hay ningún documento que guardar como, se mostrará un error.

■ **Nombre:** Exportar

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere exportar el archivo que está usando en ese momento.
2. El sistema presenta el sistema de directorios del sistema.

3. El usuario indica el directorio en el cual quiere exportar el archivo.
4. El sistema le pide al usuario que le ponga nombre al documento.
5. El sistema le pregunta al usuario con que extensión quiere guardar el archivo. Dependiendo de la extensión se guardará de una manera o de otra.
 - 5.1. Extensión .txt o .csv. En este caso el archivo se guardará con las columnas separadas por puntos y comas. La parte decimal de los números estará marcado con un punto. Las fechas tendran el siguiente formato: DD/MM/YYYY.
 - 5.2. Extensión .pdf. En este caso el sistema creará un archivo con extensión .pdf, el cual después no podrá ser importado a la aplicación.
6. El sistema informará que se ha exportado el archivo correctamente.

■ **Posibles errores:**

1. Si el archivo tiene el mismo nombre que otro previamente guardado, el sistema le preguntará si quiere sobrescribirlo o guardarlo con otro nombre.
2. Si no hay ningún documento que exportar, se mostrará un error.

■ **Nombre:** Crear Documento

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere crear un documento
2. El sistema le pregunta al usuario si quiere definir las dimensiones de este documento. En caso de que no quiera, el sistema creará un documento con una hoja, con dimensiones de 64x64. Posteriormente iremos directamente al **punto 4**
3. El usuario indica las dimensiones de las hojas que quiere crear y el número de hojas que quiere crear.
4. El sistema indica que se ha creado el documento correctamente y informa al usuario.

■ **Posibles errores:**

1. Si el archivo tiene el mismo nombre que otro previamente guardado, el sistema le preguntará si quiere sobrescribirlo o guardarlo con otro nombre.

- **Nombre:** Crear hoja

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica que quiere crear una nueva hoja.
2. El sistema le pregunta al usuario el nombre de la hoja que quiere crear.
3. El usuario indica el nombre de la hoja.
4. En caso de que el usuario no defina ningun nombre, se le pondrá uno por defecto.
5. El sistema pregunta al usuario el número de filas y columnas que quiere que tenga la nueva hoja. En caso de que el usuario no quiera definir las medidas de la hoja, estas seran de 64x64, el nombre se pondrá por defecto y posteriormente se irá al **punto 8**.
6. El usuario indica las dimensiones de la hoja que se quiere crear.
7. El usuario indica el nombre de la hoja que quiere crear.
8. El sistema indica que se ha creado la hoja correctamente.

- **Posibles errores:**

1. Si la hoja tiene el mismo nombre que otra ya existente se informará del error.

- **Nombre:** Eliminar hoja

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica que quiere eliminar una hoja.
2. El sistema envia un mensaje informando que los datos no se podran recuperar i se asegura que el usuario lo quiere eliminar.
3. Si el usuario indica que efectivamente lo que quiere eliminar, se procede a eliminar la hoja deseada.

- **Posibles errores:**

1. Si la hoja que se indica no existe en el conjunto de hojas, el sistema lo notificará.

1.1.2. Funcionalidad de gestión de Hoja

- **Nombre:** Añadir fila

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica que quiere añadir una fila y dónde la quiere añadir.
2. El sistema añade una fila en la posición indicada con el mismo número de columnas que el resto.

- **Posibles errores:**

1. Si la posición de la fila a añadir no es correcta, el sistema lo notificará.

- **Nombre:** Añadir columna

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica que quiere añadir una columna y dónde la quiere añadir.
2. El sistema añade una columna en la posición indicada con el mismo número de filas que el resto.

- **Posibles errores:**

1. Si la posición de la columna a añadir no es correcta, el sistema lo notificará.

- **Nombre:** Eliminar fila

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica que quiere eliminar una fila determinada.
2. El sistema indica que los datos eliminados no se podrán recuperar y se asegura de que el usuario la quiere eliminar.
3. El usuario indica que efectivamente quiere eliminar la fila.
4. El sistema elimina la fila indicada.

■ **Posibles errores:**

1. Si la fila contiene alguna celda que referencia a otra, se perderá la referencia y se avisará de que esta se pierde

■ **Nombre:** Eliminar columna

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere eliminar una columna determinada.
2. El sistema indica que los datos eliminados no se podrán recuperar y se asegura de que el usuario la quiere eliminar.
3. El usuario indica que efectivamente quiere eliminar la columna.
4. El sistema elimina la columna indicada.

■ **Posibles errores:**

1. Si la columna contiene alguna celda que referencia a otra, se perderá la referencia y se avisará de que esta se pierde.

■ **Nombre:** Seleccionar bloque

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario selecciona un conjunto de celdas.
2. El sistema guarda este conjunto de celdas como un bloque hasta que se seleccione uno nuevo.

■ **Posibles errores:**

1. Si el usuario selecciona un bloque con alguna celda incorrecta el sistema devuelve un mensaje de error.

■ **Nombre:** Copiar contenido del bloque

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere copiar el contenido del bloque seleccionado.
2. El sistema copia los datos del bloque y los guarda.

■ **Posibles errores:**

1. Si ya había datos guardados, estos se borran y se copian los nuevos.

■ **Nombre:** Mover contenido del bloque

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere mover el contenido del bloque seleccionado.
2. El sistema copia los datos del bloque y borra el bloque.
3. El usuario selecciona el bloque al que quiere mover los datos.
4. El sistema pega los datos previamente copiados al nuevo bloque seleccionado.

■ **Posibles errores:**

1. Si el usuario intenta mover los datos a una posición incorrecta aparecerá un mensaje de error.

■ **Nombre:** Modificar contenido del bloque

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere modificar el contenido del bloque seleccionado.
2. El usuario introduce el valor que quiere aplicar al bloque.
3. El sistema modifica el contenido del bloque aplicando a todas sus celdas el valor introducido.

■ **Posibles errores:**

1. Si el usuario intenta mover los datos a una posición incorrecta aparecerá un mensaje de error.

■ **Nombre:** Ordenación

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario selecciona un bloque a ordenar.

2. El usuario especifica la columna de referencia para la ordenación e indica que orden quiere (creciente, decreciente...).
3. El sistema ordena los valores del bloque de la manera indicada.

■ **Posibles errores:**

1. Si el usuario indica una columna de referencia la cual no contiene todo celdas tipo numéricas o todo tipo celdas tipo texto, se informará del error.

■ **Nombre:** Búsqueda de elementos

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere hacer una búsqueda en un bloque seleccionado.
2. El usuario introduce el valor a buscar.

■ **Posibles errores:**

1. Si el sistema no encuentra el valor mostrará un mensaje conforme no se ha encontrado ninguna coincidencia.

■ **Nombre:** Búsqueda y reemplazamiento de elementos

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica que quiere hacer una búsqueda en un bloque seleccionado.
2. El usuario introduce el valor a buscar e indica el valor por el que lo quiere sustituir.
3. El sistema cambia todas las celdas con el valor buscado y se sustituye por el valor para cambiar.

■ **Posibles errores:**

1.1.3. Operaciones

- **Nombre:** Truncamiento
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica a que bloque quiere aplicar la función.
 2. El sistema truncara todas las celdas de tipo numéricas del bloque seleccionado.
- **Posibles errores:**

- **Nombre:** Conversiones
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica a que bloque quiere aplicar la función.
 2. El usuario escoge de que medida a que medida quiere hacer la conversión, las conversiones posibles son las siguientes:
 - 2.1. Metros-Kilómetros y viceversa.
 - 2.2. Metros-Centímetros y viceversa.
 - 2.3. Metros-Inches y viceversa
 - 2.4. Kilómetros-Centímetros y viceversa
 3. El sistema aplicará la conversión al bloque seleccionado.
- **Posibles errores:**
 1. Si alguna celda no tiene valor, o su valor no es un número decimal, ese valor no cambiará.
- **Nombre:** Operaciones aritméticas
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario escoge la operación a aplicar entre las siguientes:
 - 1.1. Suma.
 - 1.2. Restar.
 - 1.3. División
 - 1.4. Multiplicación

2. El usuario indica con qué bloques quiere operar(puede variar según la operación a aplicar) y indicará dónde se guardará el resultado .
3. El sistema aplicará la operación correspondiente a los bloques seleccionados.

■ **Posibles errores:**

1. Si alguna celda de los bloques seleccionados no tiene valor, o su valor no es numérico, se mostrará un mensaje de error.
2. Si el bloque destino tenía alguna información almacenada esta se sobrescribirá.

● **Nombre:** Extracción

● **Actor:** Usuario

● **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El sistema le muestra al usuario las diferentes opciones para extraer.
 - 2.1. Día
 - 2.2. Mes
 - 2.3. Año
3. El usuario escoge una de las opciones presentadas.
4. El usuario indica donde quiere guardar el resultado de esta función.
5. El sistema aplica la función con los valores del bloque escogido y aplicando la opción deseada. Deja el resultado en la posición indicada.

● **Posibles errores:**

1. Si las celdas del bloque no son fechas, el sistema devolverá un mensaje de error.

● **Nombre:** Dia de la semana

● **Actor:** Usuario

● **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El usuario indica donde quiere guardar el resultado de esta función.
3. El sistema aplica la función con los valores del bloque escogido y aplicando la opción deseada. Deja el resultado en la posición indicada.

- **Posibles errores:**

1. Si las celdas del bloque no son fechas, el sistema devolverá un mensaje de error.

- **Nombre:** Reemplazamiento

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El sistema le muestra al usuario las diferentes opciones de reemplazamiento.
 - 2.1. Mayúsculas a minúsculas
 - 2.2. Minúsculas a mayúsculas
 - 2.3. Sólo la primera letra a mayúscula
3. El usuario escoge una de las opciones presentadas.
4. El usuario indica donde quiere guardar el resultado de esta función.
5. El sistema aplica la función con los valores del bloque escogido y aplicando la opción deseada. Deja el resultado en la posición indicada.

- **Posibles errores:**

1. Si las celdas del bloque no son de texto, el sistema devolverá un mensaje de error.

- **Nombre:** Consultar longitud

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El usuario indica donde quiere guardar el resultado de esta función.
3. El sistema aplica la función con los valores del bloque escogido y deja el resultado en la posición indicada.

- **Posibles errores:**

1. Si las celdas del bloque no son de texto, el sistema devolverá un mensaje de error.

- **Nombre:** Calcular la media

- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica el bloque en el que se desea aplicar la función.
 2. El usuario indica donde quiere guardar el resultado de esta función.
 3. El sistema aplica la función con los valores del bloque escogido y deja el resultado en la posición indicada.
- **Posibles errores:**
 1. Si las celdas del bloque no son numéricas, el sistema devolverá un mensaje de error.

- **Nombre:** Calcular la mediana
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica el bloque en el que se desea aplicar la función.
 2. El usuario indica donde quiere guardar el resultado de esta función.
 3. El sistema aplica la función con los valores del bloque escogido y deja el resultado en la posición indicada.
- **Posibles errores:**
 1. Si las celdas del bloque no son numéricas, el sistema devolverá un mensaje de error.

- **Nombre:** Calcular la variancia
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica el bloque en el que se desea aplicar la función.
 2. El usuario indica donde quiere guardar el resultado de esta función.
 3. El sistema aplica la función con los valores del bloque escogido y deja el resultado en la posición indicada.
- **Posibles errores:**
 1. Si las celdas del bloque no son numéricas, el sistema devolverá un mensaje de error.

- **Nombre:** Calcular la covariancia
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica el bloque en el que se desea aplicar la función.
 2. El usuario indica donde quiere guardar el resultado de esta función.
 3. El sistema aplica la función con los valores del bloque escogido y deja el resultado en la posición indicada.
- **Posibles errores:**
 1. Si las celdas del bloque no son numéricas, el sistema devolverá un mensaje de error.

- **Nombre:** Calcular la desviación estándar
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica el bloque en el que se desea aplicar la función.
 2. El usuario indica donde quiere guardar el resultado de esta función.
 3. El sistema aplica la función con los valores del bloque escogido y deja el resultado en la posición indicada.
- **Posibles errores:**
 1. Si las celdas del bloque no son numéricas, el sistema devolverá un mensaje de error.

- **Nombre:** Calcular el coeficiente de Pearson
- **Actor:** Usuario
- **Comportamiento:**
 1. El usuario indica el bloque en el que se desea aplicar la función.
 2. El usuario indica donde quiere guardar el resultado de esta función.
 3. El sistema aplica la función con los valores del bloque escogido y deja el resultado en la posición indicada.
- **Posibles errores:**
 1. Si las celdas del bloque no son numéricas, el sistema devolverá un mensaje de error.

1.1.4. Opcionales

- **Nombre:** Concatenar texto

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El usuario indica donde quiere guardar el resultado de esta función.
3. El sistema concatenará el texto de las celdas del bloque escogido y deja el resultado en la posición indicada.

- **Posibles errores:**

1. Si las celdas del bloque no son de tipo texto, el sistema devolverá un mensaje de error.

- **Nombre:** Eliminar duplicados

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El usuario introduce el valor del que desea eliminar duplicados.
3. El usuario indica donde quiere guardar el resultado de esta función.
4. El sistema buscará ese valor y borrará las celdas, excluyendo la primera, con ese valor. Deja el resultado en la posición indicada.
5. El sistema mostrará el número de duplicados que ha eliminado.

- **Posibles errores:**

1. Si el valor introducido no se encuentra en ninguna o en una única celda, el sistema devolverá un mensaje de error.

- **Nombre:** Fijar un bloque como no editable

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El sistema bloqueará las celdas en ese bloque como no editables hasta que se indique lo contrario.

- **Posibles errores:**

1. Si el bloque seleccionado o alguna celda de ese bloque ya estaba fijada como no editable, el sistema devolverá un mensaje de error.

- **Nombre:** Fijar un bloque como editable

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El sistema desbloqueará las celdas en ese bloque, dejándolas editables hasta que se indique lo contrario.

- **Posibles errores:**

1. Si el bloque seleccionado o alguna celda de ese bloque ya estaba fijada como editable, el sistema devolverá un mensaje de error.

- **Nombre:** Insertar gráficas

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El sistema le muestra al usuario los diferentes tipos de gráficas que puede escoger.
3. El usuario escoge un tipo de gráfica.
4. El sistema introduce una gráfica conforme los valores del bloque seleccionado y el tipo de gráfica escogido.

- **Posibles errores:**

1. Si el bloque seleccionado no es de celdas numéricas, el sistema devolverá un mensaje de error.

- **Nombre:** Split

- **Actor:** Usuario

- **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El usuario indica donde quiere guardar el resultado de esta función.
3. El sistema separa el texto de cada celda del bloque por espacios y deja el resultado en la posición indicada.

■ **Posibles errores:**

1. Si las celdas del bloque no son de tipo texto, el sistema devolverá un mensaje de error.

■ **Nombre:** Trim

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El usuario indica donde quiere guardar el resultado de esta función y si desea referenciarlo.
3. El sistema elimina los espacios del texto de cada celda del bloque y deja el resultado en la posición indicada.

■ **Posibles errores:**

1. Si las celdas del bloque no son de tipo texto, el sistema devolverá un mensaje de error.

■ **Nombre:** If statement

■ **Actor:** Usuario

■ **Comportamiento:**

1. El usuario indica el bloque en el que se desea aplicar la función.
2. El sistema le muestra al usuario las diferentes opciones para aplicar esta función.
3. El usuario escoge una de las opciones presentadas.
4. El usuario indica donde quiere guardar el resultado de esta función y si desea referenciarlo.
5. El sistema aplica la función con la opción elegida y deja el resultado en la posición indicada.

■ **Posibles errores:**

1. Si se intenta comparar celdas de diferentes tipos, el sistema devolverá un mensaje de error.

1.1.5. Referencias

En este subapartado explicamos que son las celdas referenciadas y que funciones pueden activar esta funcionalidad.

En esta entrega las referencias no las encadenamos, es decir, una celda no puede ser el resultado de la división de dos celdas y la suma de otras. La idea para las siguientes entregas es poder componer funciones.

Una celda referenciada es aquella que su valor depende de un conjunto no vacío de otras celdas. Las funciones que pueden activar esta funcionalidad son las siguientes.

1. Obligatorias.
 - 1.1. Truncar
 - 1.2. Convertir
 - 1.3. Sumar
 - 1.4. Restar
 - 1.5. Multiplicar
 - 1.6. Dividir
 - 1.7. Extraer
 - 1.8. Día de la semana
 - 1.9. Media
 - 1.10. Mediana
 - 1.11. Variación
 - 1.12. Standard Error
 - 1.13. Covariancia
 - 1.14. Coeficiente de Pearson
1. Opcionales.
 - 1.1. Concatenar texto
 - 1.2. Split
 - 1.3. Insertar gráficas

2. Modelo conceptual de los datos

En esta sección se explica el modelo que se ha usado en este proyecto, seguido del UML y una descripción de cada clase.

2.1. Descripción de las clases

2.1.1. Clase Documento

La clase Documento es la que se encargará de gestionar los contenidos del fichero que con la capa de persistencia serán leídos o guardados. Contendrá el conjunto de hojas y permitirá la gestión de las mismas.

2.1.2. Clase Hoja

La clase Hoja estará definida por un conjunto de celdas dispuestas en una tabla, que será donde se podrán observar todos los contenidos de las celdas en su posición correspondiente. Cuando se crea tiene un número determinado de filas y columnas que puede ser modificado por el usuario.

2.1.3. Clase Bloque

La clase Bloque se trata de un conjunto de celdas que pertenecen a una Hoja y que será donde se podrán manipular los datos para poder realizar las operaciones y funciones del programa. Solo se pueden seleccionar conjuntos rectangulares como mucho igual de grandes que el tamaño de la hoja y como mínimo de una sola celda.

2.1.4. Clase Celda

La clase Celda son unidades que pueden ser de cuatro tipos diferentes: Numérica, Textual, Fechas o Referenciada. Cuando se crea la hoja, también se crean todas las celdas (vacías) en función de las filas y columnas definidas. Las celdas también pueden hacer referencias a otras, ya sea de manera directa o mediante operaciones que la incluyan. Cuando una celda que es referenciada se modifica, también se actualizan aquellas que la referencian.

3. Estructuras de datos

En esta sección se documentan las diferentes estructuras de datos usadas en cada clase con una breve justificación y explicación de porque se han elegido estas estructuras de datos.

3.1. Documento

En clase **Documento**, necesitabamos una estructura de datos para almacenar las diferentes **Hojas**. Esta no podía ser immutable ya que un usuario puede crear y eliminar hojas en cualquier momento. Después de saber todo esto decidimos usar la librería **Vector**, una librería nativa de Java.

3.2. Hoja

Este caso es similar al comentado anteriormente, nos encontramos en una clase donde tenemos que almacenar un conjunto de celdas. En esta clase también decidimos usar la clase **Vector** ya que en cualquier momento se pueden insertar filas, columnas, etc. En esta clase también se guarda un **Bloque** para posteriormente ir haciendo operaciones con él que vaya indicando el usuario.

3.3. Celda

El caso de la clase **Celda** es diferente. En esta clase, necesitabamos saber de alguna manera si esta celda referenciaba a otras, o por otra parte, si esta celda era parte de una referencia de otras celdas. Para solucionar este problema, cada celda tiene un vector que guarda las celdas las cuales son referenciadas por ella misma. De nuevo, usamos **Vector** ya que se pueden eliminar y añadir referencias durante la ejecución del programa.

Por otra parte, necesitabamos saber, en caso de que esta celda sea referenciada por otras, de que manera estaba referenciada. Para solucionar esta cuestión, decidimos implementar un **Map** (estructura de datos nativa de Java). El *key* de este map es el nombre de la función a la cual pertenece. El *value* del map es un vector de las celdas las cuales forman parte de la operación. A continuación hay un ejemplo visual de como quedaria:

$$SUM(C1, C2, C3)$$

La operación de ejemplo nos indica que la celda la cual sea referenciada por esta operación, será el resultado de sumar las celdas *C1*, *C2* Y *C3*.

3.4. Bloque

La clase **Bloque** funciona diferente a las demás. Esto se debe a que una vez seleccionado un bloque, éste es immutable. Por tanto decidimos utilizar un **Array 2D** de celdas para implementarla.

4. Librerías externas utilizadas

Para este proyecto, hemos usado diferentes librerías externas para facilitarnos la implementación de éste. A continuación se listan las librerías y se explica para que las hemos usado.

4.1. Digidemic UnitOf

Esta librería la hemos usado para facilitar las conversiones entre unidades de medidas. El link de github donde podemos encontrar esta librería es el siguiente: [Digidemic UnitOf](#)

4.2. Knowm XChart

Esta librería se usará para próximas entregas para poder generar diferentes gráficas en la hoja de cálculo. El link de github donde podemos encontrar esta librería es el siguiente: [XChart](#)

5. Relación de clases implementadas

Document.java → Omar
DomainControler.java → Omar
Main.java → Omar

Sheet.java → Cristina
SheetTest.java → Cristina
BlockStub.java → Cristina

Cell.java → Priyanka
NumCell.java → Priyanka
TextCell.java → Priyanka
ReferencedCell.java → Priyanka
DateCell.java → Priyanka
DateCellTest.java → Priyanka
NumCellTest.java → Priyanka
ReferencedCellTest.java → Priyanka
TextCellTest.java → Priyanka
CellTest.java → Priyanka

Block.java → Mark
BlockTest.java → Mark
SheetStub.java → Mark

DocumentTest.java → Mark and Omar
CellStub.java → Mark and Priyanka