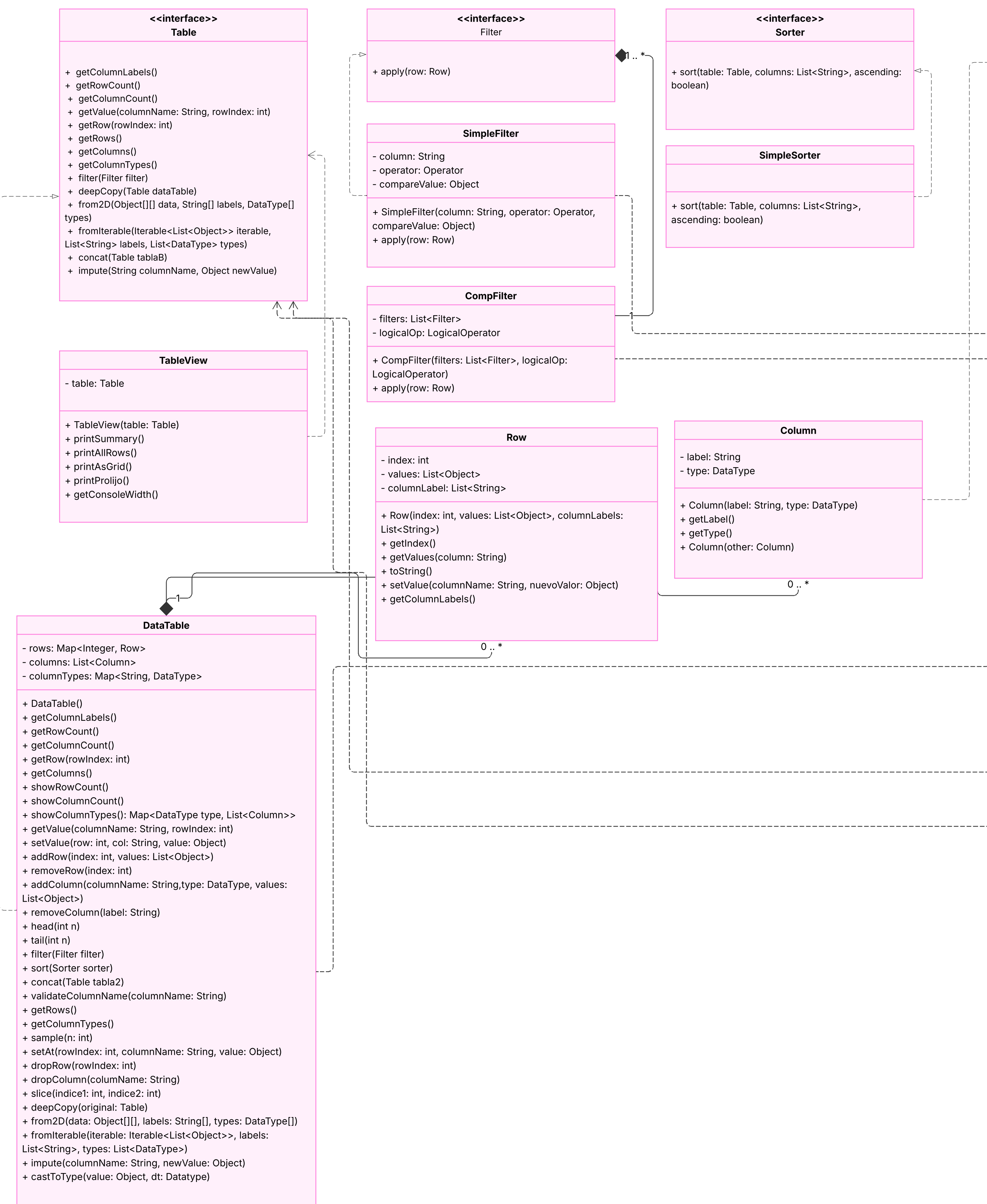# Principal

Utils

I/O

<<interface>> Table
+ getColumnLabels()
+ getRowCount()
+ getColumnCount()
+ getValue(columnName: String, rowIndex: int)
+ getRow(rowIndex: int)
+ getRows()
+ getColumns()
+ getColumnTypes()
+ filter(Filter filter)
+ deepCopy(Table dataTable)
+ from2D(Object[][] data, String[] labels, DataType[] types)
+ fromIterable(Iterable<List<Object>> iterable, List<String> labels, List<DataType> types)
+ concat(Table tablaB)
+ impute(String columnName, Object newValue)

<<interface>> Filter
+ apply(row: Row)

<<interface>> Sorter
+ sort(table: Table, columns: List<String>, ascending: boolean)

<<Enum>> DataType
+ values: INTEGER, FLOAT, DOUBLE, BOOLEAN, STRING

NA
- value: Object
- type: DataType

+ NA()
+ toString()
+ equals(obj: Object)
+ hashCode()

<<Enum>> LogicalOperator
+ values: AND, OR, NOT

SimpleFilter
- column: String
- operator: Operator
- compareValue: Object

+ SimpleFilter(column: String, operator: Operator, compareValue: Object)
+ apply(row: Row)

SimpleSorter
+ sort(table: Table, columns: List<String>, ascending: boolean)

OperationTimer
- startTime: long
- endTime: long

+ start()
+ stop()
+ getElapsedTime()
+ getElapsedSeconds()
+ getFormattedElapsedTime()

<<Enum>> Operator
+ values: EQ, EN, GT, LT, GE, LE

TableView
- table: Table

+ TableView(table: Table)
+ printSummary()
+ printAllRows()
+ printAsGrid()
+ printProlijo()
+ getConsoleWidth()

CompFilter
- filters: List<Filter>
- logicalOp: LogicalOperator

+ CompFilter(filters: List<Filter>, logicalOp: LogicalOperator)
+ apply(row: Row)

Row
- index: int
- values: List<Object>
- columnLabel: List<String>

+ Row(index: int, values: List<Object>, columnLabels: List<String>)
+ getIndex()
+ getValues(column: String)
+ toString()
+ setValue(columnName: String, nuevoValor: Object)
+ getColumnLabels()

Column
- label: String
- type: DataType

+ Column(label: String, type: DataType)
+ getLabel()
+ getType()
+ Column(other: Column)

<<interface>> TableReader
+ read(path: String, delimiter: char, hasHeader: boolean): Table

<<interface>> TableWriter
+ write(table: Table, path: String, delimiter: char, hasHeader: boolean): void

DataTable
- rows: Map<Integer, Row>
- columns: List<Column>
- columnTypes: Map<String, DataType>

+ DataTable()
+ getColumnLabels()
+ getRowCount()
+ getColumnCount()
+ getRow(rowIndex: int)
+ getColumns()
+ showRowCount()
+ showColumnCount()
+ showColumnTypes(): Map<DataType type, List<Column>>
+ getValue(columnName: String, rowIndex: int)
+ setValue(row: int, col: String, value: Object)
+ addRow(index: int, values: List<Object>)
+ removeRow(index: int)
+ addColumn(columnName: String,type: DataType, values: List<Object>)
+ removeColumn(label: String)
+ head(int n)
+ tail(int n)
+ filter(Filter filter)
+ sort(Sorter sorter)
+ concat(Table tabla2)
+ validateColumnName(columnName: String)
+ getRows()
+ getColumnTypes()
+ sample(n: int)
+ setAt(rowIndex: int, columnName: String, value: Object)
+ dropRow(rowIndex: int)
+ dropColumn(columnName: String)
+ slice(indice1: int, indice2: int)
+ deepCopy(original: Table)
+ from2D(data: Object[][], labels: String[], types: DataType[])
+ fromIterable(iterable: Iterable<List<Object>>, labels: List<String>, types: List<DataType>)
+ impute(columnName: String, newValue: Object)
+ castToType(value: Object, dt: Datatype)

CsvReader
+ read(...)

CsvWriter
+ write(...)