4. Modelo de calidad de datos

DQ Problems	DQ Dimension	DQ Factor	DQ Metric	DQ method	Applied DQ method
	ID: D4_Unicidad **Name:** Unicidad **Description:** Evalúa si los registros en un conjunto de datos son únicos y no presentan duplicaciones no deseadas **Suggested by** = {RN2, RN5}	**ID:** **Name:** No duplicación **Description:** Porcentaje de datos que no están duplicados en forma exacta **Represents** = {RN2, RN5}	**ID:** M_duplicate_ratio **Name:** duplicate_ratio **Description:** Da el porcentaje de valores duplicados. **Influenced by** = {RN2, RN5} **Granularity:** Columna **Result domain** = [0...1]	**ID:** metodo_duplicate_ratio **Name:** met_duplicate_ratio **Description:** Implementa la metrica duplicate_ratio sobre un atributo dado. **Uses** = {RN2, RN5} **Input data types:** String **Output data types:** Float **Algorithm:** "def duplicate_ratio(data, column): total = len(data) duplicated = data.duplicated(subset=[column]).sum() return duplicated / total"	**ID:** MA_duplicate_ratio **Type:** Medición **Descriptinon:** Dados los datos y el atributo, calcula el porcentaje de valores duplicados en ese atributo. **AppliedTo:** Atributos ⟪ISBN⟫, ⟪AuthorID⟫, ⟪PublisherID⟫, ⟪User_ID⟫ de la tabla NL_Books. Atributos ⟪ID⟫ de la tabla NL_USERS."

DQ Problems	DQ Dimension	DQ Factor	DQ Metric	DQ method	Applied DQ method
P11	**ID:** D2_Completitud **Name:** Completitud **Description:** Indica si el sistema de información contiene toda la información de interés. **Suggested by** = {RN1, RN2, RN5, RQ3, RQ7, F2}	**ID:** F2_Dens **Name:** Densidad **Description:** Indica cuanta información se tiene y cuanta falta sobre las entidades del sistema de información. **Represents** = {RN1, RN2, RN5, RQ3, RQ7, F2}	**ID:** M_contar_nulls **Name:** contar_nulls **Description:** Calcula el porcentaje de entradas vacias en una columna **Influenced by** = {RN1, RN5, RQ3, RQ7, F2} **Granularity:** Columna **Result domain** = {0, 1}	**ID:** metodo_contar_nulls **Name:** met_contar_nulls **Description:** Implementa la metrica contar_nulls. **Uses** = {RN1, RN5, RQ3, RQ7, F2} **Input data types:** String **Output data types:** Float **Algorithm:** "def porcentaje_nulos(atributo): nulos = atributo.isnull() contador_nulos = nulos.sum() porcentaje = contador_nulos / len(atributo) return porcentaje"	**ID:** MA_contar_nulls **Type:** Medición **Descriptinon:** Dado un atributo de una tabla, calcula el porcentaje de entradas vacias. **AppliedTo:** Cualquier atributo de cualquier tabla.

			ID: M_check_RN1
			Name: check_RN1
			Description: Da el porcentaje de entradas de la tabla que tienen al menos un campo vacio entre los atributos isbn, titulo, autor y editor.
			Influenced by = {RN1, RN2, RQ3, RQ7, F2}
			Granularity: Conjunto de columnas
			Result domain = [0...1]
				ID: metodo_check_RN1
				Name: met_check_RN1
				Description: Implementa la metrica Check_RN1 sobre una tabla dada.
				Uses = {RN1, RN2, RQ3, RQ7, F2}
				Input data types: String
				Output data types: Float
				Algorithm:
				"def Check_RN1(datos):
				# Seleccionar solo las columnas relevantes (isbn, titulo, autor, editor)
				datos_relevantes = datos[['isbn', 'titulo', 'autor', 'editor']]
				# Verificar cuántas filas tienen al menos un campo NULL entre estas columnas
				incompletos = datos_relevantes.isnull().any(axis=1).sum()
				# Calcular la proporción de filas incompletas
				return incompletos / len(datos)"
					ID: MA_check_RN1
					Type: Medición
					Descriptinon: Dada una tabla, calcula el porcentaje de entradas que tienen al menos un campo vacio entre los atributos isbn, titulo, autor y editor.
					AppliedTo:
					Tabla NL_Books.

DQ Problems	DQ Dimension	DQ Factor	DQ Metric	DQ method	Applied DQ method
	ID: D3_Consistencia **Name:** Consistencia **Description:** Captura la satisfacción de reglas semánticas definidas sobre los datos. **Suggested by** = {RQ4, RQ6, RQ8, RQ11, F1, F3}	**ID:** **Name:** Integridad inter-relación **Description:** Captura la satisfacción de reglas entre atributos de tablas distintas. **Represents** = {RQ4, RQ6, RQ8, RQ11, F1, F3}	**ID:** M_consistencia_ratings **Name:** concistencia_ratings **Description:** Verifica que la cantidad de ratings entre las distintas tablas sea coherente. **Influenced by** = {RQ8} **Granularity:** conjunto de columnas **Result domain** = {0, 1}	**ID:** metodo_consistencia_ratings **Name:** met_consistencia_ratings **Description:** Implementa la metrica concistencia_ratings contando la cantidad de ocurrencias de un ISBN valido en la tabla NL_ratings y lo compara con lo declarado en el atributo rating_count de NL_Books. **Uses** = {RQ8} **Input data types:** String **Output data types:** Boolean **Algorithm:** "def función consistencia_ratings(libros, valid_isbn, ratings): libros_validos = libros[valid_isbn==True] for libro in libros_validos: isbn = libro['isbn'] cantidad = (ratings['isbn'] == isbn).sum() contados[isbn] = cantidad rating_esperado = libros_validos['rating_counts'] return rating_esperado==contados"	**ID:** MA_consistencia_ratings **Type:** Agregación **Descriptinon:** Dada una tabla de reviews, una tabla de libros (ambas con el campo isbn) y una lista que indiquen si son validos o no, determina si hay congruencia entre lo registrado en ambas tablas sobre los ratings. **AppliedTo:** Conjunto de atributos (NL_ratings.ISBN, NL_books.ISBN, NL_books.Rating_Count)
			ID: M_consistencia_fechas **Name:** concistencia_fechas **Description:** Verifica que la fecha de un rating sea posterior a la fecha de publicación del libro. **Influenced by** = {RQ6, RQ11, F1} **Granularity:** Conjunto de columnas **Result domain** = {0, 1}	**ID:** metodo_consistencia_fechas **Name:** met_consistencia_fechas **Description:** Implementa la metrica concistencia_fechas comparando las fechas de review_time de la tabla NL_reviews y la de PublisherDate de NL_Books. **Uses** = {RQ6, RQ11, F1} **Input data types:** String **Output data types:** Boolean **Algorithm:** "def consistencia_fechas(NL_reviews, NL_books): result = lista vacia for (i ; i< len(NL_reviews); i++): isbn = NL_reviews['isbn'][i] fila_libro = fila en NL_books donde NL_books['isbn']==isbn obtener los review_time y PublishedDate de las filas correspondientes result[i] = review_time >= PublishedDate: return resut"	**ID:** MA_consistencia_fechas **Type:** Medición **Descriptinon:** Dada una tabla de reviewsy una tabla de libros (ambas con el campo isbn), determina si la fecha de la review es posterior a la publicación del libro. **AppliedTo:** Conjunto de atributos ⟪NL_Reviews.review_time, NL_Books.PublisherDate⟫
			ID: M_missing_rating_books **Name:** missing_rating_books **Description:** Para ratings sobre libros, controla la existenia de estos en la base de datos. **Influenced by** = {RQ4, RQ8, F3} **Granularity:** Celda **Result domain** = {0, 1}	**ID:** metodo_missing_rating_books **Name:** met_missing_rating_books **Description:** Implementa la metrica check_rating_books para ver si los libros indicados en los ratings existen. **Uses** = {RQ4, RQ8, F3} **Input data types:** String **Output data types:** Boolean **Algorithm:** "SELECT COUNT(*) FROM NL_Ratings WHERE NOT EXISTS(SELECT * FROM NL_Books WHERE NL_Ratings.ISBN = NL_Books.ISBN)"	**ID:** MA_missing_rating_books **Type:** Medición **Descriptinon:** Verifica que el libro en NL_Ratings exista en NL_Books mediante una consulta SQL. **AppliedTo:** Conjunto de atributos ⟪ISBN⟫ en las tablas NL_Books y NL_Ratings

DQ Problems	DQ Dimension	DQ Factor	DQ Metric	DQ method	Applied DQ method

P1, P2, P5, P6, P8, P14.	**ID:** D1_Exactitud
	Name: Exactitud
	Description: Concierne a la correctitud y la precisión con que los datos del mundo real son representados en un sistema de información.
	Suggested by = {RN1, RN2, RN4, RQ2, RQ3, RQ6, RQ11, F1}
		ID: F1_ExactSint
		Name: Exactitud sintáctica
		Description: Indica que tan libre de errores sintácticos están los datos.
		Represents = {RN1, RN2, RQ3, RN4, RQ2}
			ID: M_check_ISBN
			Name: check_ISBN
			Description: Controla si el valor es un ISBN valido.
			Influenced by = {RN1, RN2, RQ3}
			Granularity: Celda
			Result domain = {0, 1}
				ID: metodo_check_ISBN
				Name: met_check_ISBN
				Description: Implementa la métrica Check_ISBN teniendo en cuenta la estructura de un código ISBN.
				Uses = {RN1, RN2, RQ3}
				Input data types: String
				Output data types: Boolean
				Algorithm:
				"def Check_ISBN(codigo):
				size = len(codigo)
				
				# Chequea para el caso de un ISBN10
				if (size==10):
				verifica que los primeros 9 caracteres sean digitos
				verifica que el ultimo caracter sea un digito o una "X"
				verifica que el digito verificador sea correcto
				
				# Chequea para el caso de un ISBN10
				if (size==13):
				verifica que todos los caracteres sean digitos
				verifica que el digito verificador sea correcto"
					ID: MA_check_ISBN
					Type: Medición
					Descriptinon: Utiliza el algoritmo de calculo de ISBN para verificar la validez del dato.
					AppliedTo:
					Atributo ⟪ISBN⟫ de la tabla NL_Books

			ID: M_check_edades **Name:** check_edades **Description:** Controla si la edad es valida. **Influenced by** = {RN4, RQ2} **Granularity:** Celda **Result domain** = {0, 1}	**ID:** metodo_check_edades **Name:** met_check_edades **Description:** Implementa la metrica Check_edades verificando si la edad tiene formato correcto y es un valor razonable. **Uses** = {RN4, RQ2} **Input data types:** String **Output data types:** Boolean **Algorithm:** IsNumeric(edad) AND edad > 0 AND edad < 100	**ID:** MA_check_edades **Type:** Medición **Descriptinon:** Verifica que tenga un valor numerico entre 1 y 99 **AppliedTo:** Atributo ⟪Age⟫ de la tabla NL_Users
			ID: M_check_price **Name:** check_price **Description:** Controla si la precio es valido. **Influenced by** = {RN3} **Granularity**: Celda **Result domain** = {0, 1}	**ID:** metodo_check_price **Name:** met_check_price **Description:** Implementa la metrica check_price verificando si el valor tiene formato correcto **Uses** = {RN3} **Input data types:** String **Output data types:** Boolean **Algorithm:** IsNumeric(price) AND price >= 0	**ID:** MA_check_price **Type:** Medición **Descriptinon:** Verifica que tenga un valor numerico positivo con una consulta SQL **AppliedTo:** Atributo ⟪Price⟫ de la tabla NL_Books
			ID: M_duplicated_authors **Name:** duplicated_authors **Description:** Indica cuantas tuplas nombran mas de 1 autor. **Influenced by** = {RN1, RQ3} **Granularity**: Columna **Result domain** = [0..1]	**ID:** metodo_duplicated_authors **Name:** met_duplicated_authors **Description:** Implementa la metrica duplicated_author para obtener el porcentaje de tuplas que indican mas de 1 autor **Uses** = {RN1, RQ3} **Input data types:** String **Output data types:** Double **Algorithm:** DEFINE @tuplasDuplicadas = SELECT COUNT(*) FROM NL_Authors WHERE AuthorID LIKE '%,%' DEFINE @tuplasTotales = SELECT COUNT(*) FROM NL_Authors SELECT @tuplasDuplicadas / @tuplasTotales	**ID:** MA_duplicated_authors **Type:** Medición **Descriptinon:** Divide las tuplas que indican mas de 1 autor sobre las tuplas totales **AppliedTo:** Atributo ⟪AuthorID⟫ en la tabla NL_Books

DQ Problems	DQ Dimension	DQ Factor	DQ Metric	DQ method	Applied DQ method
P1, P2, P5, P6, P8, P14.	**ID:** D1_Exactitud **Name:** Exactitud **Description:** Concierne a la correctitud y la precisión con que los datos del mundo real son representados en un sistema de información. **Suggested by** = {RN1, RN2, RN4, RQ2, RQ3, RQ6, RQ11, F1}	**ID:** F3_Precision **Name:** Exactitud precisión **Description:** Indica que tan detallados son los datos. **Represents** = {RQ6, RQ11, F1}	**ID:** M_date_format **Name:** date_format **Description:** Controla el formato de fecha de todas las columnas tipo fecha. **Influenced by** = {RQ6, RQ11, F1} **Granularity**: Celda **Result domain** = {0, 1}	**ID:** metodo_date_format **Name:** met_date_format **Description:** Implementa la metrica date_format para controlar que el formato de fecha sea correcto **Uses** = {RQ6, RQ11, F1} **Input data types:** String **Output data types:** Boolean **Algorithm:** TRY_CONVERT(DATE,date,103)	**ID:** MA_date_format **Type:** Medición **Descriptinon:** Verifica que el campo PublisherDate en Books y el campo review_time en Ratings respeten el formato correcto de fecha dd/mm/yyyy **AppliedTo:** Atributo ⟪PublisherDate⟫ en la tabla NL_Books y atributo ⟪review_time⟫ en al tabla NL_Ratings