

**PROP: 4.1**

# **DISEÑO DE UN TECLADO**

**Versión 1.1**

Gabriel Carrillo:  
Alejandro del Amo:  
Josep Mercadal:

gabriel.carrillo  
alejandro.del.amo.gonzalez  
josep.mercadal



## **Índice:**

Índice:	3
Enunciado Ampliado:	4
Diagrama de Casos de Uso:	6
Descripción Diagrama de Casos de Uso:	7
Diagrama de Clases del Modelo:	22
Descripción Diagrama de Clases:	23

## **Enunciado Ampliado:**

El proyecto trata sobre el cálculo del diseño y de la disposición de un teclado para un alfabeto concreto con el fin de optimizar la velocidad de tecleo en el mismo.

La velocidad de tecleo es inversamente proporcional al esfuerzo necesario para teclear, por lo tanto, en cuanto menos esfuerzo, más velocidad.

El esfuerzo depende de la distancia entre pares de caracteres.

Por ello se requiere menos tiempo y esfuerzo para teclear dos teclas adyacentes que dos teclas en lados opuestos del teclado. Así pues, la disposición del teclado resultante será más óptimo en cuantas más teclas adyacentes haya.

Para calcular las adyacencias más frecuentes en un idioma concreto, se analizan una serie de textos en ese mismo idioma.

Así pues el usuario tiene dos opciones:

1.- Escoger uno de los 3 idiomas que el programa ofrece por defecto, que son:

Español

Inglés

Catalán

Una vez escogido, el programa muestra un conjunto de textos preestablecidos en el mismo idioma, entre los que el usuario debe seleccionar, al menos, uno para que se lleve a cabo el análisis.

2.- Definir un idioma él mismo. En este caso, se empieza por darle un nombre que no exista ya entre los idiomas del sistema. Una vez nombrado, se pasa a especificar los símbolos o letras que tiene ese alfabeto. Una vez definidos los símbolos, se pasa a introducir los textos que se quiere que sean analizados. Estos textos pueden guardarse o eliminarse del sistema en cualquier momento.

En este punto, se calculan unas estadísticas de los textos escogido anteriormente que indican los pares de letras y las letras individuales más usadas. Estos datos son empleados por los algoritmos pertinentes para calcular la disposición óptima del teclado.

Una vez obtenida la disposición óptima del teclado, si el usuario no está satisfecho con el resultado, puede recalcular la disposición de un nuevo teclado seleccionando otros textos o, si lo prefiere, puede intercambiar los símbolos o letras de dos posiciones entre sí.

Cuando se está satisfecho, se pueden guardar los resultados obtenidos.

El usuario puede cambiar el número de lados de las teclas y la disposición del teclado.

El número de lados de las teclas, que por defecto es de 4 (teclas rectangulares/convencionales), puede ser también de 6 (teclas hexagonales). Se debe tener en cuenta que cuanto mayor sea el número de lados, mayor será el número de teclas adyacentes entre sí.

La disposición del teclado, que por defecto es rectangular, puede ser también hexagonal o con forma de dos hexágonos juntos (uno para cada mano). En los teclados distinguiremos dos tipos de teclas: *fijas* y *variables*.

Las teclas *fijas* son aquellas que el sistema no tiene en cuenta a la hora de calcular el teclado óptimo, ya que el resultado no depende de ellas. El motivo por el que se tienen en consideración es que no se puede prescindir de ellas en ningún idioma, como son por ejemplo el espacio, el punto y la coma.

Por lo tanto, las teclas *variables* son todas aquellas posiciones del teclado a las que el sistema debe asignar un símbolo o letra.

Como funcionalidad opcional añadiremos que el usuario pueda dibujar sobre una matriz de un tamaño deseado de  $N \times M$ , la forma que quiera que tenga el teclado.

Para dibujar, el usuario irá haciendo “clicks” con el ratón sobre las casillas de la matriz que desee que se resalten. Estas casillas formarán los bordes del teclado (sin incluirlas). Es por esto por lo que el dibujo deberá ser *cerrado*.

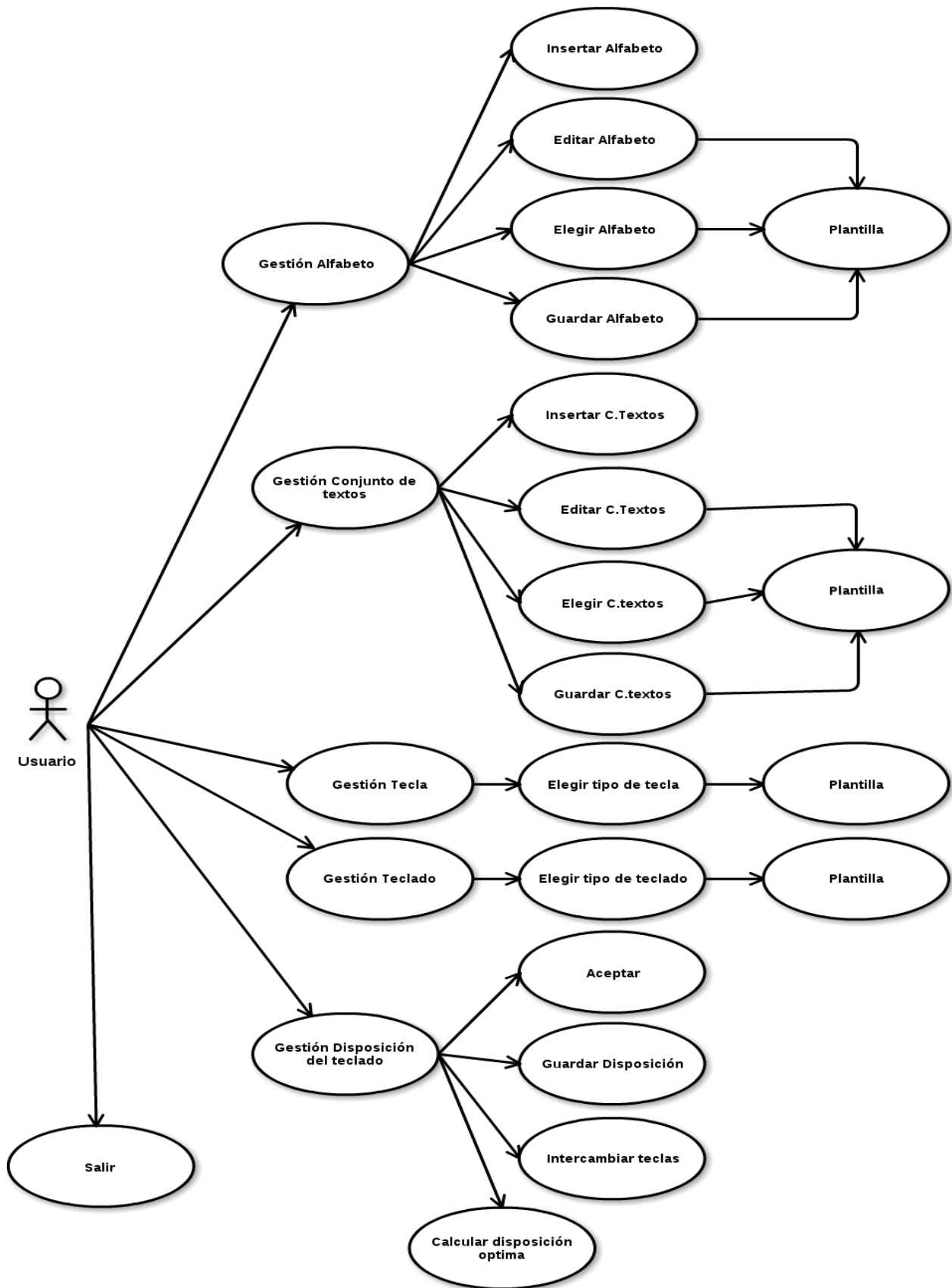
Entendemos por dibujo *cerrado* todo aquel que tenga una variación de la forma de una O. Por el contrario, un dibujo *abierto* es aquel que tenga cualquier variación de la forma de una U (está abierto por arriba).

Si el dibujo no es lo suficientemente grande como para incluir todos los símbolos o letras del idioma seleccionado, se avisará al usuario para que modifique el dibujo.

En el caso de que el dibujo sea más grande y sobren S espacios a los que no se pueda asignar ningún símbolo o letra, el sistema pedirá al usuario que marque S casillas internas al dibujo como teclas *fijas*.

Por todo lo expuesto anteriormente, con el teclado óptimo que se obtiene, se puede escribir de forma rápida y eficiente en el idioma escogido.

# Diagrama de Casos de Uso:



# Descripción Diagrama de Casos de Uso:

Caso de uso: **Insertar alfabeto.**

Actores primarios: El usuario del perfil

Activación: El usuario accede a la aplicación con la intención de insertar un nuevo alfabeto.

Escenario principal:

Usuario	Sistema
1- El usuario clics sobre “insertar alfabeto”	
	2- El sistema le pide al usuario que inserte el alfabeto que quiere
3- El usuario introduce el alfabeto que quiere	
	4 - El sistema registra el alfabeto dado por el usuario en el sistema

Posibles errores:

---

Caso de uso: **Editar alfabeto.**

Actores primarios: El usuario del perfil

Activación: El usuario accede a la aplicación con la intención de modificar uno de los alfabetos.

Escenario principal:

Usuario	Sistema
1- El usuario clics sobre “editar alfabeto”	
	2- El sistema le pide al usuario que seleccione el alfabeto que quiere editar
3- El usuario edita el alfabeto seleccionado	
	4 - El sistema registra el alfabeto editado por el usuario en el sistema

Posibles errores:

---



Caso de uso: **Elegir alfabeto.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario abre la aplicación con la intención de escoger un alfabeto.

Escenario principal:

Usuario	Sistema
1- El usuario clics sobre “escoger alfabeto”	
	2- El sistema pide al usuario que escoja uno de los alfabetos que ofrece.
3- El usuario escoge uno de los alfabetos y acepta la selección.	
	4 - El sistema guarda la selección.

Posibles errores:

---

Caso de uso: **Guardar alfabeto.**

Actores primarios: El usuario de la aplicación.

Activación: Cuando el usuario ha introducido un nuevo alfabeto, tiene la opción de guardarlo para próximos usos.

Escenario principal:

Usuario	Sistema
1- El usuario clicca sobre “guardar alfabeto”	
	2- El sistema guarda el alfabeto introducido por el usuario.

Posibles errores:

---

**Caso de uso: Insertar Conjunto de Textos.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario decide introducir él mismo un conjunto de textos.

Escenario principal:

Usuario	Sistema
1- El usuario clicla sobre “insertar conjunto de textos”	
	2- El sistema pide al usuario que asigne un título al texto que desea introducir.
3- El usuario introduce el título y acepta	
	4- El sistema pide al usuario que introduzca el texto deseado.
5- El usuario introduce el texto y acepta	
	6- El sistema guarda el nuevo texto a analizar. 7- El sistema pregunta al usuario si quiere insertar otro texto
8- El usuario selecciona si quiere insertar otro texto	
	9- En caso afirmativo, el sistema vuelve al punto 2. En caso negativo, el sistema sale del caso de uso

Posibles errores:

---

**Caso de uso: Editar Conjunto de Textos**

Actores primarios: El usuario de la aplicación

Activación: El usuario intenta modificar alguna parte de un texto

Escenario principal:

Usuario	Sistema
1- El usuario clics sobre "Editar conjunto de textos"	
	2- El sistema pide al usuario que seleccione cual es el texto que quiere modificar
3- El usuario selecciona el texto que él quiere modificar	
	4- El sistema saca por pantalla el texto y da la opción de modificar al usuario
5- El usuario modifica el texto que había seleccionado	
	6 - El sistema guarda las modificaciones hechas en el texto 7- El sistema pregunta al usuario si quiere modificar otro texto
8- El usuario selecciona si quiere modificar algún texto más	
	9- En caso afirmativo, el sistema vuelve al punto 2 y en caso negativo sale del caso de uso

Posibles errores

---

**Caso de uso: Elegir Conjunto de Textos.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario abre la aplicación y no tiene ningún texto escogido para definir el idioma

Escenario principal:

Usuario	Sistema
1- El usuario clicla sobre “escoger conjunto de textos”	
	2- El sistema le pide al usuario que seleccione al menos uno de los textos que ofrece.
3- El usuario selecciona al menos uno de los textos y acepta la selección.	
	4 - El sistema guarda la selección.

Posibles errores:

---

Caso de uso: **Guardar Conjunto de Textos.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario ha abierto la aplicación y no tiene ningún texto escogido para definir el idioma

Escenario principal:

Usuario	Sistema
1- El usuario clicca sobre “guardar conjunto de textos”	
	2- El sistema le pide al usuario que seleccione los textos a guardar.
3- El usuario selecciona los textos que desea guardar y acepta la selección.	
	4 - El sistema guarda la selección.

Posibles errores:

---

Caso de uso: **Elegir Tipo de Tecla.**

Actores primarios: El usuario de la aplicación.

Activación: Cuando el usuario tiene que escoger la forma que quiere que tengan las teclas del nuevo teclado.

Escenario principal:

Usuario	Sistema
1- El usuario clicca sobre “tipo de tecla”	
	2- El sistema le pide al usuario que seleccione entre la forma rectangular y la hexagonal.
3- El usuario selecciona la tecla que quiere y acepta la selección.	
	4 - El sistema guarda la selección del usuario para la nueva disposición de las teclas.

Posibles errores:

---

Caso de uso: **Elegir Tipo de Teclado.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario tiene que elegir la forma de su teclado.

Escenario principal:

Usuario	Sistema
1- El usuario clics sobre “elegir tipo de teclado”	
	2- El sistema ofrece al usuario las posibles formas del teclado con una plantilla.
3- El usuario selecciona uno de los tipos y acepta la selección.	
	4 - El sistema guarda la selección.

Posibles errores:

---



Caso de uso: **Aceptar disposición.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario ha introducido el alfabeto y los textos y la aplicación le da una disposición óptima, que quiere aceptar.

Escenario principal:

Usuario	Sistema
	1- El sistema le da al usuario una disposición óptima del teclat a través de los datos introducidos por él
2- Si al usuario le gusta la nueva disposición del teclado, la aceptara y sinó puede volver al principio para cambiar los datos o intercambiar las teclas que no le gusten	

Posibles errores:

---

Caso de uso: **Guardar Disposición.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario ha introducido un alfabeto y unos textos y la aplicación le da una disposición óptima, que quiere guardar para otra ocasión.

Escenario principal:

Usuario	Sistema
	1- El sistema le da al usuario una disposición óptima del teclado a través de los datos introducidos
2- El usuario le gusta la disposición del teclado y la quiere guardar	
	3- El sistema le pide que le asigne un nombre a la disposición del teclado
4- El usuario introduce un nombre para la disposición	
	5 - El sistema guarda la disposición y el nombre que ha introducido el usuario

Posibles errores:

-Que el usuario intente introducir un nombre que ya esté en la base de datos. El sistema ha de sacar por pantalla un mensaje de “Nombre ya utilizado”, y volver a preguntar que nombre quiere ponerle a la disposición

Caso de uso: **Intercambiar Teclas.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario quiere intercambiar dos teclas de una disposición.

Escenario principal:

Usuario	Sistema
1- El usuario clica sobre "Intercambiar teclas"	
	2- El sistema pide al usuario que selecciona la disposición sobre la que quiere intercambiar las teclas
3- El usuario selecciona la disposición sobre la que quiere hacer el intercambio	
	4- El sistema muestra por pantalla la disposición que ha seleccionado el usuario 5- El sistema pregunta al usuario cuales son las teclas que quiere intercambiar
6 - El usuario selecciona las 2 teclas que quiere intercambiar	
	7- El sistema saca por pantalla la nueva disposición

Posibles errores:

---

Caso de uso: **Calcular disposición óptima**

Actores primarios: El usuario del perfil

Activación: El usuario decide obtener la disposición óptima tras especificar los datos necesarios

Escenario principal:

Usuario	Sistema
1- El usuario clicla sobre “calcular disposición óptima”	
	2- El sistema ejecuta los algoritmos pertinentes sobre las estadísticas obtenidas para dar la disposición óptima 3- El sistema saca por pantalla la disposición óptima

Posibles errores

---

Caso de uso: **Salir.**

Actores primarios: El usuario de la aplicación.

Activación: El usuario quiere cerrar la aplicación.

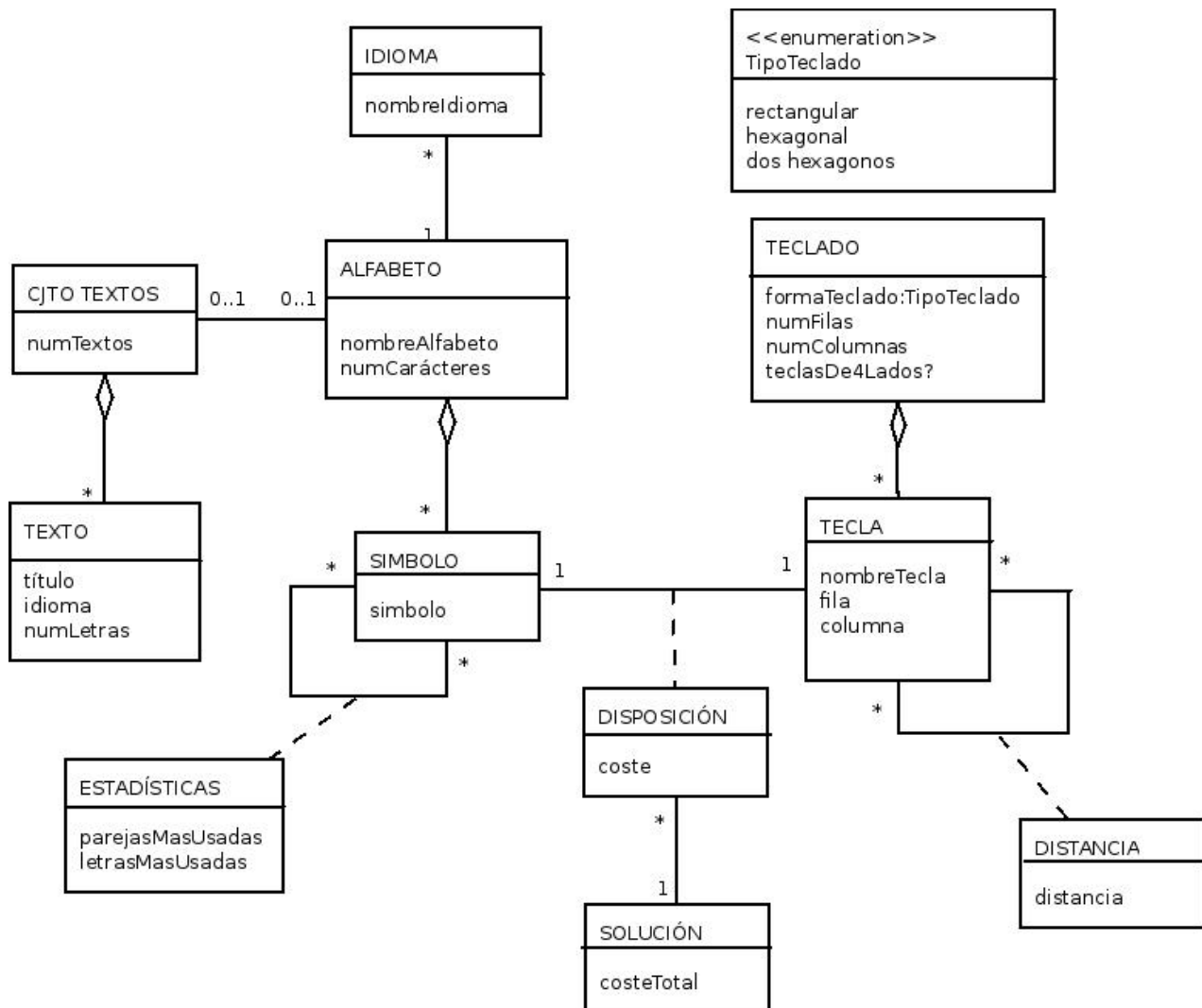
Escenario principal:

<b>Usuario</b>	<b>Sistema</b>
1- El usuario clica sobre “Salir”	
	2- El sistema finalitza y cierra la aplicación

Posibles errores:

---

# Diagrama de Clases del Modelo:



## Restricciones de Integridad:

1.- Claves Externas: (Idioma, nombreIdioma), (Alfabeto, nombreAlfabeto), (Símbolo, símbolo), (Texto, título), (Posición, fila+columna), (Tecla, nombreTecla), (Teclado, forma+numFilas+numColumnas)

## 2.- Restricciones Textuales:

**RT1:** El teclado debe tener el mismo número de teclas variables que de caracteres tiene el alfabeto escogido.

# Descripción Diagrama de Clases:

## **Idioma:**

Explicación: La clase define el idioma de al menos uno de los textos.

Cardinalidad: Hay 3 idiomas por defecto. El usuario puede añadir más.

Atributos: nombreIdioma

- nombreIdioma (String): Identificador de la clase [*Català, Castellano, ...*].

Asociaciones: La clase tiene una asociación directa.

- Puesto que cada idioma emplea un único alfabeto, la asociación directa con la clase Alfabeto tiene cardinalidad 1.

## **Alfabeto:**

Explicación: La clase representa el conjunto de símbolos o letras que definen un alfabeto concreto.

Cardinalidad: La cardinalidad de la clase es como mínimo de 3 puesto que por defecto hay 3 alfabetos. El usuario puede insertar cualquier alfabeto que desee.

Atributos: nombreAlfabeto, numCaracteres

- nombreAlfabeto (String): Identificador de la clase [*Cirilico, Armenio, ...*].
- numCaracteres (Integer ): Cantidad de caracteres que contiene un alfabeto [*27, 29, ...*].

Asociaciones: La clase tiene dos asociaciones directas y contiene una agregación.

- Puesto que un alfabeto puede ser usado por muchos idiomas, la asociación directa con la clase Idioma tiene cardinalidad \*.
- Puesto que un alfabeto puede ser definido por un conjunto de textos, la asociación con la clase ConjuntoDeTextos tiene cardinalidad 0..1.
- Puesto que un alfabeto es un conjunto de símbolos, la relación de agregación con la clase Simbolo tiene cardinalidad \*.

**Símbolo:**

Explicación: La clase representa la letra o símbolo que forma parte de un alfabeto.

Cardinalidad: La cardinalidad de la clase viene definida por el número de caracteres de la clase Alfabeto.

Atributos: simbolo.

- simbolo (Char): Identificador de la clase  $[a, b, \dots]$ .

Asociaciones: La clase tiene una asociación recursiva, una directa y forma parte de una agregación.

- Puesto que un símbolo tiene unas estadísticas con un conjunto de símbolos, la asociación recursiva Estadísticas tiene cardinalidad \*.
- Puesto que a un símbolo se le pueden calcular las estadísticas con un conjunto de símbolos, la asociación recursiva Estadísticas tiene cardinalidad \*.
- Puesto que un símbolo tiene una disposición en una tecla, la asociación directa con la clase Tecla llamada Disposición tiene cardinalidad 1.
- Puesto que una agregación de símbolos definen un alfabeto, se ha hecho una relación de agregación con la clase Alfabeto.

**Estadísticas:**

Explicación: Calcula las letras y las parejas de letras más frecuentes de los textos seleccionados por el usuario.

Cardinalidad: La cardinalidad es 1 puesto que se sacan unas únicas estadísticas de un mismo conjunto de textos.

Atributos: parejasMasUsadas, letrasMasUsadas

- parejasMasUsadas(Pair<Char, Char>): Pares de letras más usadas  $[ab, cp, \dots]$ .
- letrasMasUsadas(Char): Letras más usadas  $[e, r, \dots]$ .

Asociaciones: La clase es una clase asociativa entre dos símbolos.

- Puesto que unas estadísticas se llevan a cabo sobre la combinación de todos los símbolos con los demás, la asociación recursiva Simbolo tiene multiplicidad \* en ambos lados.



**Conjunto de textos:**

Explicación: Define el conjunto de textos seleccionados por el usuario.

Cardinalidad: La cardinalidad es 1, y está formada por tantos textos como el usuario seleccione para definir su idioma.

Atributos: numTextos

- numTextos (Integer): indica el número de textos registrados en el sistema  
Ej: [1, 3, ...].

Asociaciones: La clase tiene una asociación directa y contiene una agregación.

- Puesto que un conjunto de textos puede definir un alfabeto, la asociación directa con la clase Alfabeto tiene cardinalidad 0..1.
- Puesto que un conjunto de textos es un conjunto de textos, la relación de agregación con la clase Texto tiene cardinalidad \*.

**Texto:**

Explicación: Un texto tiene un título, está escrito en un idioma y contiene un conjunto de letras.

Cardinalidad: La cardinalidad de la clase es como mínimo de 3 puesto que por defecto hay 3 textos, uno para cada idioma que hay por defecto. El usuario puede insertar cualquier texto que desee.

Atributos: titulo, idioma, numLetras

- titulo (String): Identificador de la clase [*La Regenta, El ingenioso hidalgo don Quijote de la Mancha, ...*].
- idioma (String): Idioma en el que está escrito el texto [*Castellano, English, ...*].
- numLetras(Integer): Número de letras que contiene el texto [*91.328, 1.618.098*].

Asociaciones: La clase forma parte de una agregación.

- Puesto que una agregación de textos definen un conjunto de textos, se ha hecho una relación de agregación con la clase ConjuntoDeTextos.

## **Teclado:**

Explicación: Un teclado contiene un conjunto de teclas. El usuario puede elegir, entre las opciones que se ofrecen, la forma que quiera darle al teclado. Esto variará la disposición del teclado. El usuario puede variar el número de filas y columnas respetando la forma del teclado seleccionado.

Cardinalidad: La cardinalidad de la clase es como mínimo 3 puesto que por defecto hay 3 tipos de teclado.

Atributos: tipoForma, numFilas, numColumnas

- formaTeclado (TipoTeclado) : Nombre que define la forma que tiene el teclado [*rectangular, hexagonal, ...*].
- numFilas (Integer): Número de filas que tiene el teclado [*5, 6, ...*].
- numColumnas (Integer): Número de columnas que tendrá el teclado [*15, 17, ...*].
- teclasDe4Lados? (Boolean): Booleano que en el caso de ser **true** indica que la tecla tiene cuatro lados. En el caso de ser **false** indica que tendrá 6 lados [*true, false, ...*].

Asociaciones: La clase contiene una agregación.

- Puesto que un teclado es un conjunto de teclas, la relación de agregación con la clase Tecla tiene cardinalidad \*.

**Tecla:**

Explicación: Una tecla representa una celda del teclado a la que se ha de asignar un símbolo. Las teclas pueden tener dos formas: rectangulares o hexagonales. La forma seleccionada variará la disposición del teclado, ya que el número de lados y las adyacencias aumentan equitativamente.

Cardinalidad: La cardinalidad de la clase viene definida por el número de caracteres del alfabeto.

Atributos: nombreTecla, fila, columna.

- nombreTecla (Char): Indica la letra que contendrá esa tecla [*'a', 'b', ...*].
- fila (Integer): Fila en la que se encuentra la tecla [*2, 3, ...*].
- columna (Integer): Columna en la que se encuentra la tecla [*2, 7, ...*].

Asociaciones: La clase tiene una asociación recursiva, una directa y forma parte de una agregación.

- Puesto que una tecla tiene unas distancias con un conjunto de símbolos, la asociación recursiva Distancia tiene cardinalidad \*.
- Puesto que a una tecla se le pueden calcular las distancias con un conjunto de teclas, la asociación recursiva Distancia tiene cardinalidad \*.
- Puesto que una tecla dispone de un símbolo, la asociación directa con la clase Simbolo llamada Disposición tiene cardinalidad 1.
- Puesto que una agregación de teclas definen un teclado, se ha hecho una relación de agregación con la clase Teclado.

**Distancia:**

Explicación: La clase representa la distancia de las posiciones entre sí.

Cardinalidad: La cardinalidad de la clase viene definida por el número de caracteres del alfabeto.

Atributos: distancia.

- distancia (Integer): Distancia entre dos teclas [*1, 5, ...*].

Asociaciones: La clase es una clase asociativa entre dos teclas.

- Puesto que una distancia se calcula sobre la combinación de todas las teclas con las demás, la asociación recursiva Distancia tiene multiplicidad \* en ambos lados.

**Disposición:**

Explicación: La clase representa la relación entre un símbolo y una posición.

Cardinalidad: La cardinalidad de la clase dependerá del número de caracteres de la clase alfabeto.

Atributos: coste

- coste (Integer): Coste que tiene hacer esta disposición [3, 4, ...].

Asociaciones: La clase tiene una asociación directa y es una clase asociativa entre un símbolo y una tecla.

- Puesto que una disposición es empleada en una solución, la asociación directa con la clase Solucion tiene cardinalidad 1.
- Puesto que una disposición se obtiene mediante la combinación de un símbolo con una tecla, la asociación Disposición tiene multiplicidad 1 en ambos lados.

**Solución:**

Explicación: La clase representa la disposición definitiva del teclado.

Cardinalidad: La cardinalidad de la clase es 1 puesto que se obtiene una única solución óptima.

Atributos: costeTotal

- costeTotal (Integer): Coste total de la disposición [51, 30, ...].

Asociaciones: La clase tiene una asociación directa.

- Puesto que una solución emplea un conjunto de disposiciones, la asociación directa con la clase Disposicion tiene cardinalidad \*.