

---

# Traductor de Pictogramas a Texto

---



Trabajo de Fin de Grado  
Curso 2017–2018

Autor

Salvador González Álvarez  
José María López Pulido

Director

Virginia Francisco Gilmartín  
Susana Bautista Blasco

Trabajo de Fin de Grado en Ingeniería del Software  
Facultad de Informática  
Universidad Complutense de Madrid



# Traductor de Pictogramas a Texto

Trabajo de Fin de Grado en Ingeniería del Software

**Autor**

Salvador González Álvarez  
José María López Pulido

**Director**

Virginia Francisco Gilmartín  
Susana Bautista Blasco

**Convocatoria:** *Febrero/Junio/Septiembre 2018*

**Calificación:** *Nota*

Trabajo de Fin de Grado en Ingeniería del Software

Facultad de Informática

Universidad Complutense de Madrid

18 de diciembre de 2018



# Autorización de difusión

El abajo firmante, matriculado en el Máster en Ingeniería en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “TITULO DEL TRABAJO”, realizado durante el curso académico CURSO bajo la dirección de DIRECTORES en el Departamento de XXXXXXXXXXXXXXXXXXXXXXXXXX, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Nombre Del Alumno

18 de diciembre de 2018



# Dedicatoria

Texto de la dedicatoria...





# Agradecimientos

Texto de los agradecimientos



# Resumen

Desde las primeras civilizaciones, el ser humano ha sentido la necesidad de comunicarse, en la actualidad esta necesidad es plasmada día a día en Internet: redes sociales, blogs... el hombre del siglo XXI, es más social y por ese motivo la necesidad de comunicarse es mas importante que nunca. No todas las personas tienen la misma capacidad de expresarse y o comunicarse, y por esta razón nacen los sistemas alternativos de comunicación.

El uso de pictogramas es uno de estos sistemas, en los últimos años hemos visto una evolución en la utilización de pictogramas en el ámbito educativo, en especial en niños con TEA (Trastorno del Espectro Autista) y trastornos del lenguaje en general como puedan ser Asperger, Síndrome de Down, parálisis cerebral, etc.

Por estos motivos hemos decidido crear un servicio web que permita al usuario traducir pictogramas a lenguaje natural. Una aplicación que facilite y ayude a la comunicación entre personas con diferentes capacidades comunicativas.

## Palabras clave

Máximo 10 palabras clave separadas por comas



# Abstract

Abstract in English.

## **Keywords**

10 keywords max., separated by commas.



# Índice

<b>1. Introduction</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Motivación . . . . .	3
1.2. Objetivos . . . . .	4
1.3. Estructura del trabajo . . . . .	4
<b>2. Estado de la Cuestión</b>	<b>5</b>
2.1. Sistemas Aumentativos y Alternativos de Comunicación . . . .	5
2.2. Pictogramas . . . . .	6
2.2.1. Sistemas pictográficos . . . . .	8
2.3. Generación de lenguaje natural . . . . .	14
2.3.1. Fases de GLN . . . . .	16
2.4. Servicios Web . . . . .	18
2.4.1. Ventajas de los servicios Web . . . . .	20
2.4.2. Inconvenientes de los servicios Web . . . . .	20
2.5. Sistema de traducción texto-picto . . . . .	21
2.5.1. Araword . . . . .	21
2.5.2. PictoTraductor . . . . .	22
2.5.3. Pictar . . . . .	23
2.6. Metodología de trabajo . . . . .	24
2.6.1. Metodología ágil . . . . .	24
2.6.2. Control de versiones: GitHub . . . . .	27
<b>3. Herramientas</b>	<b>31</b>
3.1. Django . . . . .	31
3.1.1. ¿Qué es Django? . . . . .	31
<b>4. Conclusiones y Trabajo Futuro</b>	<b>33</b>

<b>4. Conclusions and Future Work</b>	<b>35</b>
<b>A. Título</b>	<b>37</b>
<b>B. Título</b>	<b>39</b>
<b>Bibliografía</b>	<b>41</b>



# Índice de figuras

2.1. Pictograma que representa un smarthwatch. . . . .	7
2.2. Pictograma que representa pintarse los labios. . . . .	7
2.3. Pictograma que representa una espada. . . . .	8
2.4. Figura Bliss(1). . . . .	9
2.5. Figura Bliss(2). . . . .	9
2.6. Figura Bliss(3). . . . .	10
2.7. Sistema MinSpeak. . . . .	11
2.8. Clasificación de los pictogramas SPC. . . . .	11
2.9. Diferentes conjunciones con pictogramas de Arasaac. . . . .	12
2.10. Tablero ARASAAC para actividades escolares. . . . .	13
2.11. Traducción en pictogramas de la frase Hoy vamos a la montaña	13
2.12. Ejemplo de verbos conjugados . . . . .	14
2.13. Pictogramas para diferente numero y genero . . . . .	14
2.14. Pictogramas de hoy, mañana, y pasado . . . . .	15
2.15. Ejemplo de frase con conjunción temporal . . . . .	15
2.16. Fases de generación del lenguaje. . . . .	16
2.17. Árbol representativo de la fase de macro planificación en GLN.	17
2.18. Estructura servicios Web. . . . .	19
2.19. Ejemplo artículos Araword. . . . .	21
2.20. Ejemplo preposiciones Araword. . . . .	22
2.21. Ejemplo yo quiero comer pizza. . . . .	23
2.22. Ejemplo artículos PictoTraductor. . . . .	23
2.23. Ejemplo preposición PictoTraductor. . . . .	24
2.24. Frase en pictoTraductor . . . . .	24
2.25. Presentación pictar . . . . .	25
2.26. Texto traducido en pictar . . . . .	25
2.27. Ejemplo buscador y editor pictar . . . . .	26
2.28. Ejemplo de tablero Scrum . . . . .	27
2.29. Tablero agile del proyecto . . . . .	28

2.30. Ejemplo de árbol de Git . . . . .	28
2.31. Aplicación GitHub de escritorio . . . . .	29

# Índice de tablas



# Chapter 1

## Introduction

Introduction to the subject area.



# Capítulo 1

## Introducción

*“El hombre, mediante palabras, expresa su significado”*

— San Agustín

### 1.1. Motivación

La comunicación es una parte fundamental del desarrollo social y humano, pero las personas con diversidades funcionales se encuentran muchos obstáculos para una comunicación correcta. La aparición de diferentes vías de comunicación, como por ejemplo los pictogramas, ha supuesto un gran avance para lograr una comunicación efectiva en este colectivo. Pero la realidad es que la comunicación sigue estando muy limitada y en ocasiones se reduce a personas con el mismo tipo de diversidad funcional. Este es el caso de usuarios cuya vía de comunicación son los pictogramas, ya que el resto de usuarios desconoce su significado lo que produce que la comunicación quede notablemente reducida o en muchos casos sea imposible.

Esta es la razón, por la que existe la necesidad de crear una herramienta que ayude a la comunicación de los usuarios de pictogramas con su entorno, un software que sea capaz de traducir pictogramas a lenguaje natural. Esta herramienta software permitirá derribar barreras y ayudar a la inclusión social de personas que por su diversidad funcional están en riesgo de exclusión.

El objetivo de este TFG es el de crear un traductor de pictogramas a texto, los principales beneficiados serán usuarios con grandes problemas para poder comunicarse, como pueden ser: personas con parálisis cerebral, autismo o cualquier otro tipo de enfermedad de este tipo utilizar los pictogramas comunicarse de manera fluida. Además este trabajo está pensado en ayudar a esas personas a poder comunicarse en cualquier ambiente cotidiano de la vida, desde poder pedir un refresco en el bar hasta expresar su deseo de comer macarrones. Algo tan simple que para cualquier persona resulta muy fácil de hacer y que por desgracia para ellos no lo es tanto. Por ello crearemos

una aplicación que permita la interpretación (no la traducción literal) de mensajes con pictogramas a texto. Los resultados de este trabajo ayudarán a las personas que presentan problemas de inclusión debido a la necesidad de usar pictogramas para comunicarse.

## 1.2. Objetivos

El objetivo principal de este trabajo, es la creación de un traductor de pictogramas a lenguaje natural, buscando la interpretación del mensaje y no una traducción literal. Un producto que sea accesible para todos, buscando la facilidad de uso a través de una interfaz cómoda accesible e intuitiva, para de esta manera llegar al máximo publico posible, además uno de los puntos fuertes de este trabajo será la utilización de Servicios Web, esto será muy importante ya que como explicaremos en capítulos posteriores ayudará a relacionar diferentes tecnologías manteniendo su independencia unas de otras. Además de poder integrar nuestro trabajo dentro del proyecto Idylico formado entre ellos por algunos profesores de la facultad de informática y que llevan a cabo diferentes trabajos relacionados con el objetivo principal de nuestro trabajo de final de grado.

Buscando afianzar los conocimientos adquiridos durante nuestra vida universitaria, además de añadir más competencias en la informática, nos embarcamos en este último trabajo antes de salir a la vida laboral. Además al tratarse de un trabajo que posee una utilidad social que será usado por personas que lo necesitan, hemos tenido la oportunidad de poner en práctica muchas características propias de la ingeniería del software, al tener que cumplir unos requisitos y desarrollar en base a ellos nuestro trabajo. Para ello desarrollaremos un servicio web, que permita introducir un conjunto de pictogramas y devuelva al usuario, una interpretación en lenguaje natural del significado del mensaje introducido. Integraremos este servicio dentro de una aplicación web accesible para todo el mundo.

## 1.3. Estructura de la memoria

La memoria de este trabajo estará formada por un total de cuatro capítulos incluyendo esta introducción, el cuál será el primer capítulo. A continuación detallaremos el contenido de cada capítulo.

1. En el capítulo dos **Introduction**, tendremos una breve presentación del trabajo en inglés.
2. En el capítulo tres **Estado de la cuestión**, hablaremos de todos los aspectos importantes para la generación del lenguaje natural, el uso de



---

pictogramas y de SAACS. Además de introducir los Servicios web y la metodología llevada a cabo en este trabajo.

3. En el capítulo cuatro **Herramientas**, será donde se describa las diferentes herramientas utilizadas para la confección de este trabajo. Aquí se presenta una breve información sobre Django y sobre la herramienta GSON.
4. En el capítulo quinto **Trabajo individual**, se expondrá el trabajo individual realizado por la pareja encargada de realizar el trabajo.



## Capítulo 2

# Estado de la Cuestión

A lo largo de este capítulo trataremos los aspectos más importantes que han sido necesarios o forman parte de la base de este trabajo como los Sistemas Aumentativos y Alternativos de Comunicación, pictogramas...

### 2.1. Sistemas Aumentativos y Alternativos de Comunicación

La comunicación es una necesidad, y en casos de autismo o parálisis cerebral donde el lenguaje oral está gravemente limitado, la utilización de sistemas de comunicación de lenguaje no verbales, sustituyen o sirven de apoyo al lenguaje verbal.

Los Sistemas Aumentativos y Alternativos de Comunicación<sup>1</sup> (SAAC) son distintas formas de expresar el lenguaje oral, que tienen como objetivo aumentar y/o compensar los problemas de comunicación de muchas personas con dificultades para conseguir una expresión verbal funcional.

El término comunicación aumentativa describe las formas en las que se comunica el individuo cuando el uso del lenguaje natural no es suficiente, por el contrario la comunicación alternativa hace referencia a diferentes métodos o vías de comunicación que sustituyen completamente el lenguaje oral, como por ejemplo los pictogramas o la lengua de señas.

Lo que se pretende con los SAAC es conseguir una comunicación funcional, adecuada y generalizable, que permita al individuo expresarse y alcanzar una mayor integración en su entorno social. Los SAAC ayudan a mejorar la calidad de vida de las personas con problemas para comunicarse, al otorgar al individuo una vía para mejorar su comunicación. Cabe destacar que los SAAC no son usados exclusivamente por personas con alteraciones en la comunicación, sino que son utilizados por todos a diario, por ejemplo cuando

---

<sup>1</sup><http://www.arasaac.org/aac.php>

empleamos gestos, señales de tráfico, etc.

Según la American SpeechLanguage-Hearing Association(Romski y Sevcik), Lloyd y Karlan (1984) además de Carmen Basil (Basil et al., 1990) los SAAC se pueden clasificar en dos grandes grupos:

1. SAAC sin ayuda: Sistemas que carecen de soporte físico. Abarca desde el uso de mímica y gestos hasta el uso de signos manuales.
2. SAAC con ayuda: sistemas que necesitan un soporte físico independiente del emisor. Este tipo abarca desde sistemas sencillos basados en dibujos o imágenes hasta objetos reales en miniatura o los sistemas de comunicación escritos como el Braille. Están orientados a personas con problemas en el habla o con dificultades cognitivas o de aprendizaje. Dentro de los SAAC con ayuda se encuentran los pictogramas de los cuales hablaremos en la siguiente sección.

Se ha demostrado que la utilización de los SAAC tiene múltiples ventajas (Álvarez, 2013) y son indispensables para aquellas personas que en mayor o menor medida, tienen impedido el uso del lenguaje. Sus usuarios potenciales son aquellas personas que por patologías como el autismo, la parálisis cerebral, o lesiones cerebrales, están impedidos para expresarse de manera adecuada o correcta(Warrick, 2010). Además de ser un recurso necesario para estas personas se ha demostrado que tienen múltiples beneficios siendo los principales:

1. Ayudan al desarrollo de la comunicación, proporcionando estrategias el usuario.
2. Posibilitan el desarrollo personal y social ademas de fomentar las relaciones interpersonales.
3. Evitan el aislamiento, fomentando habilidades afectivo-sociales.

Cabe destacar que la complejidad de los SAAC depende del nivel cognitivo del usuario (Abril Abadín, 2010), cada persona es diferente, y por ejemplo en el uso de imágenes, el nivel de abstracción y complejidad deben adecuarse a las necesidades de cada individuo.

## 2.2. Pictogramas

Se define pictograma como un signo icónico dibujado y no lingüístico, que representa un objeto real o un significado. Se engloban dentro de los SAACS con ayuda, y se aplican a personas que no están alfabetizadas ya sea por causa de la edad o de alguna discapacidad. En la Figura 2.1 podemos observar un ejemplo de pictograma que representa un smartwatch. La



Figura 2.1: Pictograma que representa un smartwatch.



Figura 2.2: Pictograma que representa pintarse los labios.

utilización de pictogramas para la escritura o pictografía se remonta al neolítico, donde se utilizaban piedras talladas para la representación de signos de cierta semejanza con el objeto que representan. El origen de los sistemas pictográficos actuales se sitúa en 1981 con Mayer y Johnson y se caracteriza por una facilidad de interpretación.

Las características principales de los pictogramas son los siguientes:

1. Guardan relación con aquello que representan, tal y como se puede ver en el pictograma de la Figura 2.1 el cual representa un smartwatch.
2. Son elementos gráficos que combinados representan el objeto tomado como referente. Por ejemplo en el pictograma de la Figura 2.2 la cara, el lápiz de labios y la mano se juntan en un solo pictograma para representarla acción pintarse los labios.
3. La imagen debe de ser comprensible por el mayor número de personas, independientemente de la formación, idioma o discapacidad.
4. A la hora de construir un pictograma, se deben de seguir unas reglas que le permitan mantener una coherencia visual, es decir la legibilidad de un pictograma debe de ser inmediata.
5. Deben de ser sencillos y representar únicamente los elementos más importantes, evitando posibles estímulos distractores o información irre-

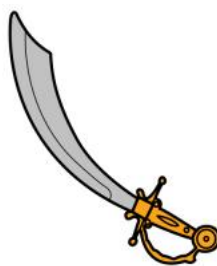


Figura 2.3: Pictograma que representa una espada.

levante. Por ejemplo, en la Figura 2.3 el objeto representado es una espada, el pictograma carece de elementos distractores como podría ser una mano que agarre la espada o un fondo que sirva de contexto.

Existen diferentes colecciones de pictogramas que vamos a analizar en la siguiente subsección.

### 2.2.1. Sistemas pictográficos

A continuación vamos a ver algunos de los principales sistemas pictográficos existentes, además de explicar en último término como se realiza la comunicación a través de pictogramas :

#### 2.2.1.1. Bliss

Bliss<sup>2</sup> fue desarrollado por Charles K. Bliss en 1949 y está formado por más de 2000 símbolos. Tiene la dificultad de que el usuario debe conocer el sistema ya que no cumple la relación con el objeto representado tal y como se puede ver en la Figura 2.4, donde los pictogramas de animal o papel, no guardan ninguna referencia con aquello que representan y dado solo el pictograma no puedes conocer que objeto representa.

El sistema pictográfico Bliss fue creado con la idea de ser un lenguaje global (McDonald, 1985). Está formado por un total de 26 letras en inglés y 28 para el castellano, además de un alfabeto de 26 símbolos básicos, la combinación de estas letras y estos símbolos básicos forman las palabras, es por esta razón que la utilización de este sistema pictográfico requiera de cierto nivel cognitivo tal y como se explica en (Hehner, 1985). Según el nivel de representación de los símbolos los pictogramas se agrupan en tres grandes grupos: pictográficos, ideográficos y arbitrarios:

1. Símbolos pictográficos: la forma del símbolo recuerda las palabras o conceptos que representan, esto lo podemos observar en el símbolo que

---

<sup>2</sup><https://www.uv.es/bellochc/logopedia/NRTLogo8.wiki?6>



Figura 2.4: Figura Bliss(1).



Figura 2.5: Figura Bliss(2).

representa hogar en la Figura 2.5, con el pictograma de la casa o la silla que recuerdan el objeto que representan.

2. Símbolos ideográficos: expresan una idea, no describen el objeto, por ejemplo en la Figura 2.5 podemos ver el símbolo de hogar, el cual tiene un corazón que significa sentimientos, que es no es algo tangible.
3. Símbolos arbitrarios: expresan un concepto abstracto y el símbolo no guarda ningún relación con el objeto representado, como podemos observar en la Figura 2.6, el significado del pictograma es un concepto abstracto que no guarda ninguna referencia con el símbolo representado.



Figura 2.6: Figura Bliss(3).

### 2.2.1.2. Minspeak

Es un sistema pictográfico, que creo por Bruce Baker en 1982<sup>3</sup> motivado por las carencias que observó en personas sin habla. Este sistema no estipula un significado unico para cada pictograma, si no que el significado viene dado por una secuencia de pictogramas, de ahí que en un sistema MinSpeak el pictograma que representa un arco iris puede significar felicidad pero en otro puede significar lluvia.

Bruce Baker se inspiro en el lenguaje Chino donde cualquier frase puede tener 12 caracteres o menos (Marín y Pérez, 2003), y de esa manera definió que su sistema que se centra en la combinación de pictogramas. De esta manera a través de la combinación de diferentes símbolos, normalmente de dos o tres, más un conjunto de reglas y patrones sirve para la representación de una palabra o frase. Como se puede observar en la Figura 2.7, combinando el símbolo que representa una casa con el de cama se obtiene el significado de habitación. Se considera un sistema muy accesible<sup>4</sup> dado que el usuario debe de aprender un número limitado de pictogramas que suelen ir desde los 40 hasta los 80 pictogramas, variando este numero dependiendo de las capacidades del usuario, las palabras ademas quedan clasificadas por grupos semánticos, es decir: verbos, adjetivos... Permite desarrollar un lenguaje a través de la construcción de frases o secuencias que son fáciles de entender, pero tiene la desventaja de que el usuario necesita recordar la secuencia de símbolos necesarios para la generación de la respuesta correcta.

### 2.2.1.3. SPC

El Sistema Pictografico de Comunicación (SPC)<sup>5</sup> fue desarrollado por Roxana Mayer-Jhonson en 1981 que buscaba a través de un diseño de pic-

<sup>3</sup><http://ares.cnice.mec.es/informes/18/contenidos/94.htm>

<sup>4</sup><https://minspeak.com/>

<sup>5</sup><http://masquemayores.com/magazine/psicologia/tipos-de-sistemas-alternativos-y-aumentativos-de-la-comunicacion-sistemas-pictograficos-de-comunicacion/>



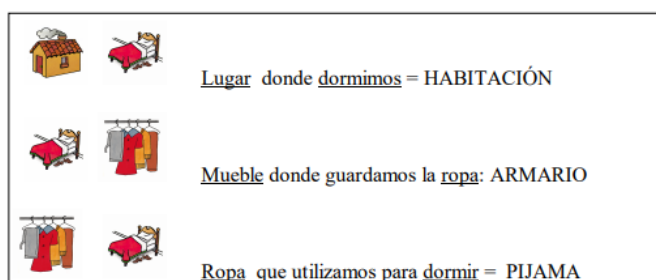


Figura 2.7: Sistema MinSpeak.



Figura 2.8: Clasificación de los pictogramas SPC.

trogramas simples, la similitud con aquello que representan. Los pictogramas en SPC están catalogados en seis grupos cada uno con un color para ayudar a la clasificación (ver en la Figura 2.8) personas en amarillo, verbos en verde, pictogramas descriptivos en azul, sustantivos en naranja, letras, números y miscelánea con fondo blanco y actos sociales como saludarse en rosa. En total el sistema SPC lo forman un conjunto de 3.000 pictogramas.

Los usuarios pueden comunicarse utilizando SPC señalando los pictogramas en tableros, o en un libro o pueden coger los pictogramas pegarlos en un panel o utilizar tablets o pantallas a modo de tablero. Por tanto la comunicación queda limitada a los pictogramas seleccionables. Además la comunicación mediante SPC carece de sintaxis lo que simplifica la construcción de frases.

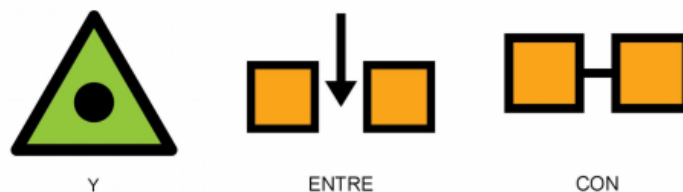


Figura 2.9: Diferentes conjunciones con pictogramas de Arasaac.

#### 2.2.1.4. ARASAAC

ARASAAC<sup>6</sup>, nace en el año 2007, financiado por el Departamento de Educación Cultura y Deporte del Gobierno de Aragón y coordinado por la Dirección General de Innovación, Equidad y Participación de dicho departamento. Este proyecto nace con el objetivo de romper las barreras en la comunicación. ARASAAC cuenta con más de 33.000 pictogramas únicos, repartidos entre pictogramas a color, en blanco y negros y con traducciones en más de quince idiomas entre ellos el castellano.

Actualmente, ARASAAC se ha convertido en un sistema de pictogramas reconocido internacionalmente, entre otros factores porque sus recursos están bajo la licencia Creative Commons.

Los pictogramas están clasificados en cinco grandes grupos: pictogramas a color, en blanco y negro, fotografías y por ultimo fotografías y vídeos en Lengua de Signos Español (Bertola López, 2018-02-15) ordenados dependiendo de su naturaleza semántica. Nos centraremos en los pictogramas a color, donde podemos encontrar diferenciación de genero y numero, además es importante destacar que existen pictogramas para algunas conjunciones y determinantes tal y como podemos ver en la imagen 2.9

Por ultimo destacar que los pictogramas de ARASAAC tienen genero: masculino, femenino y neutro además de tener número, tal y como podemos ver en la Figura 2.13, donde se muestra las diferentes variaciones tanto en genero y número para la palabra profesor.

Además opcional mente se puede añadir un marco al borde del pictograma dicho marco sirve para clasificar la palabra diferenciando entre sujetos, verbos y objetos tal y como podemos ver en la Figura ?? los sujetos no llevan borde, las acciones o verbos llevan un borde verde y los objetos o complementos directos llevan un borde amarillo.

<sup>6</sup><http://www.arasaac.org>



Figura 2.10: Ejemplo de bordes de color en pictogramas

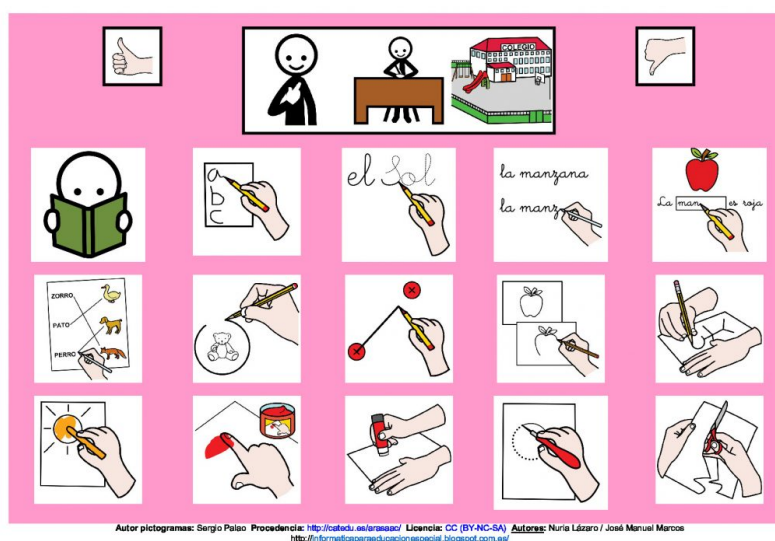


Figura 2.11: Tablero ARASAAC para actividades escolares.

### 2.2.1.5. Comunicación a través de pictogramas

Comúnmente la comunicación con pictogramas se realiza a través de tableros, de manera que el usuario señala uno a uno los pictogramas en dicho tablero hasta completar la frase. En la Figura 2.10 podemos observar un tablero de ejemplo con relación a actividades desarrolladas en el colegio, además podemos ver dos pictogramas uno que representa un pulgar hacía arriba y otro hacía abajo, que sirven para comunicar si o no respectivamente, el conjunto inferior de los pictogramas guardan relación con acciones que cualquier alumno podría encontrarse en el colegio, leer, hacer un examen, dibujar, ejercicios de caligrafía...

Una palabra puede representarse con varios pictogramas, como podemos observar en la Figura ???. La palabra "vamos" tiene diferentes pictogramas asociados, y su uso dependerá del nivel cognitivo del usuario el contexto de la frase, etc.

Los verbos carecen de conjugación como podemos observar en la Figura



Figura 2.12: Traducción en pictogramas de la frase Hoy vamos a la montaña

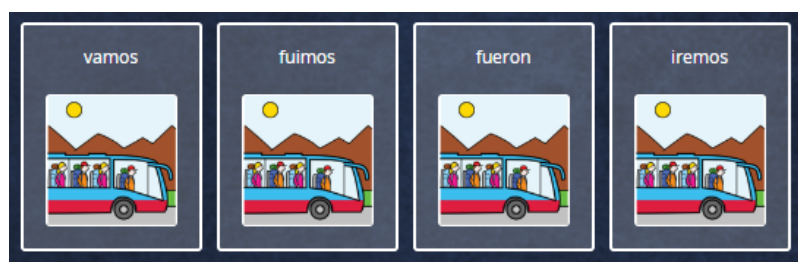


Figura 2.13: Ejemplo de verbos conjugados

2.12 donde se puede ver que no existen conjugaciones. Los tiempos verbales: presente, pasado o futuro quedan determinados por otros pictogramas, como podemos observar en la Figura 2.14 que presenta los diferentes pictogramas para hoy, mañana y pasado, estos pictogramas nos ayudan a dar tiempo verbal a la frase.

El genero y numero si se representa a través de diferentes variaciones de los pictogramas, tal y como podemos observar en la Figura 2.13 que representa los pictogramas asociados a profesor, profesora, profesores y profesoras. Las frases con pictogramas suelen tener una complejidad reducida, limitándose casi siempre a sujeto, verbo y objeto, utilizando solo palabras significativas. Por ejemplo, en la Figura 2.11, podemos observar que la preposición *a* y el determinante *la*, no tienen un pictograma como tal si no una imagen con la palabra<sup>7</sup>.

Como hemos dicho anteriormente la comunicación con pictogramas se reduce a palabras significativas, es decir palabras que por si mismas poseen significado y es por está razón por la que determinantes, conjunciones, preposiciones no suelen tener cabida dentro de la comunicación con pictogramas, aunque ciertas conjunciones sobretodo conjunciones temporales que sirven para aportar información si existen dentro de los pictogramas tal y como podemos ver en la Figura 2.15, el primer pictograma significa antes y aporta información sobre cuando debe realizarse la acción definida en el segundo pictograma, en este caso la frase representada seria ".antes comer".

<sup>7</sup>[www.pictotraductor.com](http://www.pictotraductor.com)



Figura 2.14: Pictogramas para diferente numero y genero



Figura 2.15: Pictogramas de hoy, mañana, y pasado

## 2.3. Generación de lenguaje natural

La Generación de Lenguaje Natural (GLN) es un campo de la Inteligencia Artificial para crear programas informáticos que generan lenguaje natural ya sea hablado o escrito<sup>8</sup>.

La GLN se engloba dentro de la lingüística computacional aunque engloba muchas otras áreas de estudio fuera de la informática como la lingüística o la psicología. Se busca que el programa generador se comunique igual que si de una persona se tratase (Dale et al., 2000; Vicente et al., 2015).

Para que la generación del lenguaje sea efectiva la inteligencia artificial además necesita poder procesar el lenguaje natural, es decir, poder entender y analizar el significado de aquello que se dice. Falsamente podemos pensar que lo más importante de la generación del lenguaje es la generación de un texto gramaticalmente correcto, pero lo más importante de la comunicación es que el texto generado explique aquello que se desea transmitir, por eso podemos concluir que el proceso de generación se podría resumir en la cons-

<sup>8</sup><https://searchdatacenter.techtarget.com/es/definicion/Generacion-de-lenguaje-natural-o-NLG>



Figura 2.16: Ejemplo de frase con conjunción temporal

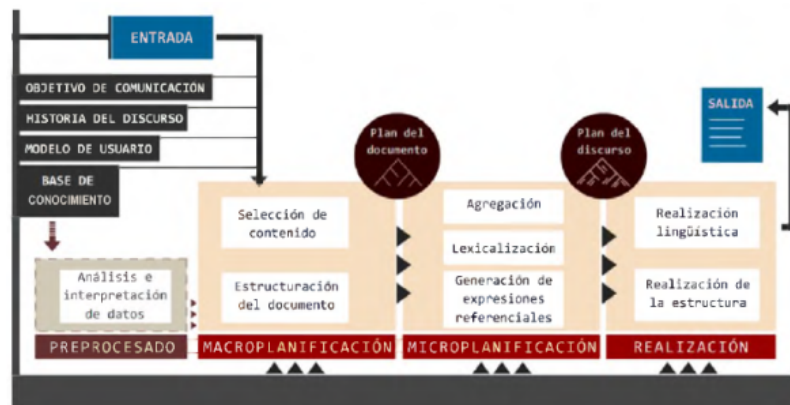


Figura 2.17: Fases de generación del lenguaje.

trucción de un mensaje en lenguaje natural que transmite un mensaje de manera clara.

Existe dos maneras de que un programa informático se comuniquen con un usuario. La primera forma es a través de mensajes estáticos definidos en el código, este acercamiento si bien es valido, genera un sistema cerrado y no flexible en el cual no cabe lugar la interpretación del lenguaje. La segunda vía, es la generación de lenguaje basada en conocimiento o *deep generation* donde es el propio sistema el encargado de generar el lenguaje a través de su conocimiento basándose en el conocimiento lingüístico del que dispone (García Ibáñez, 2004).

### 2.3.1. Fases de GLN

Durante la generación del lenguaje se puede abstraer el proceso y encapsularlo en diferentes fases que describiremos brevemente a continuación (Vicente et al., 2015):

1. Macro planificación. Etapa donde se debe seleccionar el contenido que se desea convertir a texto. Para que el texto generado sea coherente

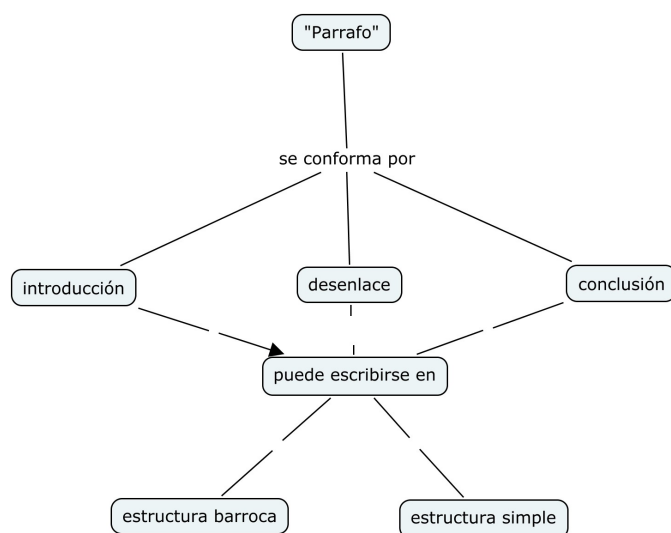


Figura 2.18: Árbol representativo de la fase de macro planificación en GLN.

es preciso estructurar el contenido del mismo en el orden correcto. La salida de esta fase es el denominado plan del documento, una estructura de datos que suele ser un árbol, donde en cada nodo se encapsula la información más importante que debe formar parte de un párrafo o de una frase, además de información de como se relaciona con el resto de nodos. En la Figura 2.17 podemos ver un ejemplo de la salida de esta etapa.

2. Micro planificación. Partiendo de la salida de la macro planificación, se determinan que partes formaran el texto resultado, agregando las estructuras que formaran parte de este, el léxico que se utilizara para expresar los conceptos y los hechos recogidos en la primera etapa y la generación de expresiones que aparecerán en el texto.
3. Realización. Se parte del resultado de la fase de micro planificación, el objetivo de esta fase es la realización de las oraciones finales que se utilizaran, así como la estructura final que tendrá el texto. La generación o realización del texto se puede dividir en dos partes realización lingüística en la que se determina la representación en palabras del texto de salida es decir se forman las oraciones, y la realización de la estructura en la que se genera el texto en el formato correcto para su visualización en un medio determinado, como puede ser HTML, o un documento Latex.

La herramienta SimpleNLG-ES es una biblioteca Java que permite generar y transformar texto. Tiene como fin producir texto comprensible para el

usuario, para ello posee tres componentes conectadas entre sí (?).

Al ser una biblioteca es necesario escribir un programa Java que haga uso de las clases de SimpleNLG-ES. Estas clases permiten especificar tanto el sujeto de una oración, como el verbo, así como el complemento directo y los complementos adicionales. Además los métodos SimpleNLG-ES pueden usarse también para indicar tanto el tiempo pasado como la forma compuesta de los verbos.

Cuando se ha decidido el contenido de la oración, SimpleNLG-ES juntará las partes de la oración de una forma gramaticalmente correcta y generará una salida. Por ejemplo, si indicamos a SimpleNLG-ES que el sujeto 'mi perro', el verbo es 'y' de complemento 'feliz', el programa sacará como respuesta: 'mi perro es feliz'. Como acabamos de ver en el ejemplo SimpleNLG-ES se encarga únicamente de asegurar que la frase creada es correcta gramaticalmente, para eso la herramienta tiene automatizadas algunas tareas para comprobar lo que los sistemas de generación de lenguaje natural necesitan (NLG):

**"Ortografía:"**

1. Insertar espacios en blanco para separar las frases.
2. Mezclar los signos de puntuación.
3. Insertar saltos de líneas entre palabras para encajar el texto.
4. Darle un formato correcto.

**"Morfología:"**

1. Genera formas conjugadas, como puede ser el género, número, tiempo verbal o persona.

**"Gramática simple:"**

1. Asegura la concordancia entre nombre-verbo.
2. Crea verbos compuestos bien formados.
3. SimpleNLG-ES une las partes definidas por el usuario para formar una estructura correcta.

Es importante recordar que SimpleNLG-ES es una librería y no una aplicación. Esto significa que no se puede ejecutar como un programa Java normal ya que no tiene ni método y módulo principal. Por tanto se necesita crear un programa diferente con clases y métodos.



### 2.3.2. Lexicon

Como todos los sistemas de procesamiento del lenguaje natural, SimpleNLG necesita tener información acerca de las palabras que va a utilizar, esta información la proporciona un lexicon, se usa dicho lexicon mediante:

1. `Lexicon lexicon = new XMLLexicon();`

Una vez esté creado el lexicon, hay que crear un `NLGFactory`, objeto que crea la estructura, y un `Realiser`, objeto que transforma las estructuras en texto, se inicializa de la siguiente forma:

1. `NLGFactory nlgFactory = new NLGFactory(lexicon);`
2. `Realiser realiser = new Realiser(lexicon);`

### 2.3.3. Verbos

Por lo general los verbos se deben especificar en infinitivo, pero también aceptará formas conjugadas de los verbos, por ejemplo:

1. `p.setVerb(".es");`

equivale a:

1. `p.setVerb("ser");`

Sin embargo, en SimpleNLG-ES se pueden conjugar los tiempos verbales: pasado, presente y futuro, por ejemplo partiendo del siguiente código:

1. `SPhraseSpec p = nlgFactory.createClause();`
2. `p.setSubject("Mi perro");`
3. `p.setVerb("ser");`
4. `p.setObject("feliz.");`

Para poder poner el verbo en tiempo pasado, habría que añadir la siguiente línea:

1. `p.setFeature(Feature.TENSE, Tense.PAST);`

Esto devolverá la frase:

1. "Mi perro fue feliz".

En cambio si queremos indicar un tiempo futuro, sería de esta manera:

1. `p.setFeature(Feature.TENSE, Tense.FUTURE);`

Devolviendo la frase:

1. "Mi perro será feliz".

Además, también se puede añadir la forma negativa al verbo, cogiendo la frase anterior sería:

1. `p.setFeature(Feature.NEGATED, true);`

Dejando la frase en:

1. "Mi perro no será feliz".

Las características TENSE y NEGATED son ejemplos de las características que se usan en una SPhraseSpec, pero hay otras más como son: MODAL, PASSIVE, PERFECT, y PROGRESSIVE.

## 2.4. Servicios Web

El WC3 (World Wide Web Consortium) define un Servicio Web como<sup>9</sup>:

*Un sistema software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable. Cuenta con una interfaz descrita en un formato procesable por un equipo informático (específicamente en WSDL), a través de la que es posible interactuar con el mismo mediante el intercambio de mensajes SOAP, típicamente transmitidos usando serialización XML sobre HTTP conjuntamente con otros estándares web.*

Un Servicio Web es un componente accesible mediante peticiones Web estándar tales como GET o POST. Normalmente se denomina servicio web a una colección de procedimientos accesibles desde Internet. La gran ventaja de los servicios web es que al acceder desde el navegador a un servicio web este nos devuelve el contenido con un formato estándar como puede ser XML.

Los servicios Web fueron creados para solucionar el problema de la interoperabilidad entre las distintas aplicaciones. En los 90, el gran objetivo era poder integrar aplicaciones distintas dentro de un mismo servicio. Estas aplicaciones pueden estar desarrolladas en muchos lenguajes de programación como pueden ser: C++, Java, etc. Y ejecutarse tanto en Unix, en un PC, o un computador mainframe. Por estas razones, no existía una forma fácil de comunicar todas las aplicaciones entre sí. Gracias al desarrollo de XML, propició poder compartir datos entre ellas a través de la red, o de Internet.

A nivel técnico, los servicios Web pueden ser dos tipos.

---

<sup>9</sup><https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211>

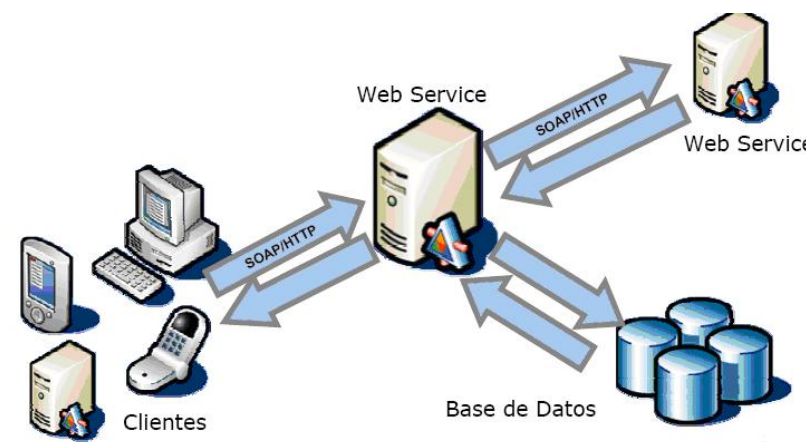


Figura 2.19: Estructura servicios Web.

- Servicios Web SOAP. Utilizan el lenguaje XML para poder comunicarse utilizando el protocolo SOAP (Simple Object Access Protocol), este lenguaje es usado tanto para definir la arquitectura como el formato de los mensajes. Estos tipos de servicios deben contener un lenguaje legible para la máquina con la descripción de operaciones ofrecidas por el servicio escrito en WSDL (Web Services Description Language), el cuál está basado también en XML con el fin de definir interfaces sintácticamente.
- Servicios Web Restfull (Representational State Transfer Web Services). Los servicios Web Restfull utilizan lenguajes muy conocidos como SML, URI, MIME, y sobre todo HTML. Además posee una infraestructura la cuál podríamos llamar "ligera", ya que consiguen utilizar herramientas mínimas para formar estos servicios. Por ello, el desarrollo de este tipo de servicios son baratos y su adaptación es poco costosa. Son más adecuados para integraciones básicas ad-hoc. Estos servicios suelen mezclarse mejor con HTTP que los basados en SOAP, ya que no necesitan mensajes XML ni definiciones WSDL.

En una arquitectura de Servicios Web se pueden distinguir tres partes fundamentales para el establecimiento de la comunicación<sup>10</sup>:

- Proveedor de servicios web. Es el encargado de enviar al publicador del servicio un fichero WSDL con la información básica de dicho servicio.
- Publicador del servicio. Una vez la información llega al publicador este descubre quien es el proveedor (protocolo WSDL), y se pone en contacto con él (protocolo SOAP).

<sup>10</sup><http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html>

- Proveedor del servicio. Acepta la petición del servicio y envía los datos estructurados en formato XML usando SOAP.

### 2.4.1. Ventajas de los servicios Web

Las tres principales ventajas de usar web son<sup>11</sup>:

- Ayudan a mejorar la conexión y operabilidad entre dos aplicaciones software sin importar sus propiedades o sino comparten plataforma de instalación.
- Favorecen los protocolos y estándares basados en texto, ya que estos facilitan acceder y comprender su contenido.
- Permiten y facilitan que software y servicios de diferentes compañías se puedan combinar formando un servicio funcional.

### 2.4.2. Inconvenientes de los servicios Web

Las tres principales desventajas de los servicios son<sup>12</sup>:

- Las transacciones que llevan consigo estos servicios están mucho menos desarrolladas que otros estándares abiertos.
- Su rendimiento es bajo comparado con otros modelos de computación distribuida, debido sobre todo a adoptar un formato basado en texto.
- Al estar muy ligados con HTTP, las medidas de seguridad que tratan de bloquear la comunicación entre programas se pueden burlar fácilmente.

## 2.5. Sistema de traducción texto-picto

En esta sección se presentan los sistemas de traducción texto a picto existentes, ya que no se ha encontrado ningún sistema que haga la traducción de picto-texto, objetivo principal de este trabajo.

### 2.5.1. Araword

Araword<sup>13</sup> es una aplicación gratuita creada por Arasacc (Portal Aragonés de la Comunicación Aumentativa y Alternativa), que permite la traducción simultanea de texto a pictograma.

Esta herramienta tiene como objetivo principal facilitar la elaboración de materiales con pictogramas. Se parece a un editor de textos, se comienza con

<sup>11</sup><https://sites.google.com/site/preyctodetics/home/servicios-web>

<sup>12</sup><http://fabioalfarocc.blogspot.com/>

<sup>13</sup><http://www.arasaac.org/software.php>



Figura 2.20: Ejemplo artículos Araword.



Figura 2.21: Ejemplo preposiciones Araword.

un documento en blanco y se van escribiendo frases en lenguaje natural se van traduciendo a pictogramas. Cada vez que se introduce una palabra en el editor, la propia herramienta es la que se encarga de traducirla a pictograma automáticamente. Cuando Las palabras que tienen varios pictogramas como significado, Araword te muestra una de esas imágenes dando la posibilidad de poder elegir otro pictograma si este no lo consideras adecuado. En Araword las preposiciones y los artículos son muy variados, diferenciando el género y número en el caso de los artículos sin traducir literalmente la palabra, como podemos ver en la (Figura 2.19). Y en las preposiciones teniendo un pictograma para cada una de ellas , como aparece en la 2.20). Por último, se puede ver en la (Figura 2.21) un ejemplo de la traducción hecha por Araword del texto.



Figura 2.22: Ejemplo yo quiero comer pizza.

### 2.5.2. PictoTraductor

PictoTraductor<sup>14</sup> es una aplicación web pensada y desarrollada para facilitar la comunicación con personas que poseen dificultades para expresarse de forma oral y que se comunican mediante pictogramas. Principalmente está dirigido a padres y profesionales, para traducir textos a pictogramas y así poder comunicarse con sus hijos o pacientes que solo entienden pictogramas y no lenguaje natural.

La interfaz de la aplicación se puede ver en la (Figura 2.24). Consta de una barra de búsqueda en la que se introduce el texto que se quiere traducir a pictogramas. A medida que se va escribiendo una palabra se va mostrando el pictograma que representa el texto introducido en la parte inferior. Las palabras que tengan más de un pictograma para poder ser traducido, la herramienta da la opción de elegir la que más guste al usuario mediante las flechas que aparecen encima del picto traducido. Para los artículos PictoTraductor deja por defecto la traducción literal de la palabra, pero como acabamos de mencionar deja la opción de poder cambiarlo a gusto del usuario, como vemos en la 2.22). En las preposiciones cada traducción tiene su pictograma, a diferencia de los artículos en PictoTraductor solo hay un pictograma para cada una de las preposiciones. 2.23).

Esta herramienta permite el registro de usuarios, personalizar la aplicación, por ejemplo, guardando sus pictogramas favoritos, fotografías o imágenes y compartir algunas de sus traducciones en las redes sociales.(Figura

<sup>14</sup><https://www.pictotraductor.com/>



Figura 2.23: Ejemplo artículos PictoTraductor.



Figura 2.24: Ejemplo preposición PictoTraductor.

2.24).

### 2.5.3. Pictar

Pictar (Martín Guerrero, 2018) es una aplicación web que permite traducir de texto a pictogramas, presentando un aspecto tal y como se muestra en la 2.25) .

El usuario introduce el texto que desea traducir y al pulsar el botón buscar se generan los pictogramas adecuados para la frase introducida, tal y como se puede ver en la (Figura 2.26) .

La búsqueda de artículos y preposiciones es muy parecida a la de PictoTraductor, poniendo en la barra de texto las palabras deseadas, una vez pinchado el botón de búsqueda aparecen los pictogramas y mediante el uso de flechas se puede ir cambiando hasta encontrar el tipo de pictograma que el usuario considere idóneo.

Esta aplicación además ofrece la posibilidad de buscar pictogramas. El usuario introduce una palabra y en la parte inferior aparecen los pictogramas



Figura 2.25: Frase en pictoTraductor

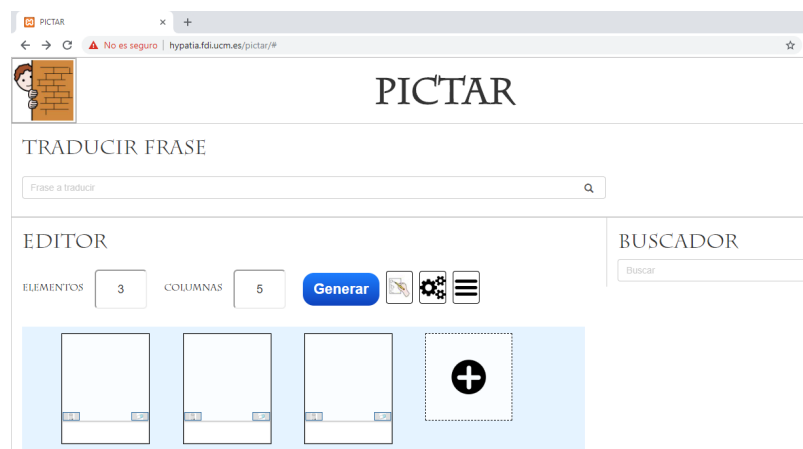


Figura 2.26: Presentación pictar

encontrados en el buscador. Una vez encontrados los pictogramas deseados, el usuario puede moldearlos a su gusto arrastrando con el ratón el pictograma a la rejilla de edición. En ella se podrá desde cambiar el color del pictograma hasta poder añadir texto debajo de ellos con su significado, como se puede ver en la (Figura2.27) .

## 2.6. Metodología de trabajo

A lo largo de la siguiente sección expondremos como hemos decidido trabajar, organización, realización de pruebas, sistema de control de versiones...

### 2.6.1. Metodología ágil

La mayor limitación que nos hemos encontrado a la hora de aplicar una metodología de trabajo es el numero de personas necesarias para su aplica-



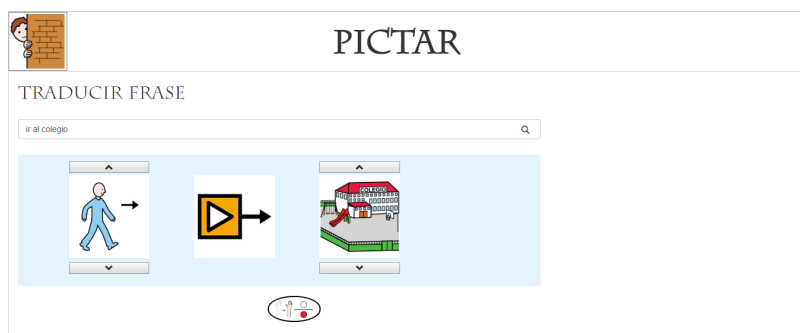


Figura 2.27: Texto traducido en pictar

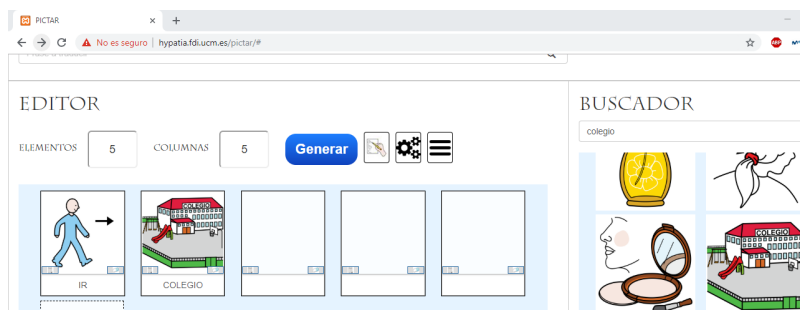


Figura 2.28: Ejemplo buscador y editor pictar

ción, por la metodologías ágiles como Scrum<sup>15</sup> necesitan de 5 a 9 personas.

Dada la naturaleza del proyecto, hemos decidido escoger ciertas y los impedimentos hemos decidido tener una visión orientada hacia las metodologías ágiles, cogiendo diferentes características de las metodologías ágiles.

- Abierto al cambio, dada la naturaleza del proyecto, el desconocimiento de su viabilidad y de ser viable el modo de desarrollarlo, por esto la visión debía de estar abierta al cambio.
- Comunicación continua, en los procesos ágiles está comunicación se realiza con el cliente, dada la naturaleza del proyecto la comunicación es con los tutores, esta comunicación ha sido constante desde el principio del proyecto con el fin de garantizar un nivel de calidad en el producto y asesoramiento en la realización.
- Adaptativa, al ser un proyecto de investigación es necesario que el proyecto se pueda adaptar fácilmente a cambios de requisitos, uso de tecnologías nuevas etc.

<sup>15</sup><https://proyectosagiles.org/que-es-scrum/>

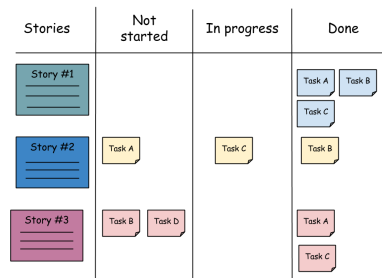


Figura 2.29: Ejemplo de tablero Scrum

- Entregas constantes, para intentar asegurar la correcta resolución del proyecto se han realizado entregas constantes tanto de memoria como de código.
- Se otorga valor al producto entrega tras entrega, cada una de las entregas anteriores no solo sirve para revisión de entregas anteriores si no también para añadir valor al proyecto ya sea mediante la adición en la memoria, al código o ambas.

Muchas de las metodologías ágiles hacen uso de un tablero, tal y como podemos observar en la Figura 2.28 para representar el estado actual del proyecto, y dado ciertas limitaciones como no trabajar en el mismo sitio, ni con el mismo horario, necesitábamos de un tablero que nos permitiera de una manera rápida saber el estado actual del trabajo, y en que está trabajando cada miembro.

En la Figura 2.29 podemos ver nuestro tablero agile, donde aparecen las siguientes columnas:

- To Do: donde están apuntadas todas las cosas que se deben hacer.
- In Progress: en esta columna están apuntadas todas las tareas que se están desarrollando además de mostrar quien la está haciendo.
- Ready For Revision: todas las tareas finalizadas por un miembro del equipo pasan a esta columna para ser revisadas por el otro.
- In Revisión: tareas que se están revisando, para validar su corrección antes de la siguiente reunión.
- Done: en esta columna se agrupan todas aquellas tareas que ya están terminadas y revisadas.

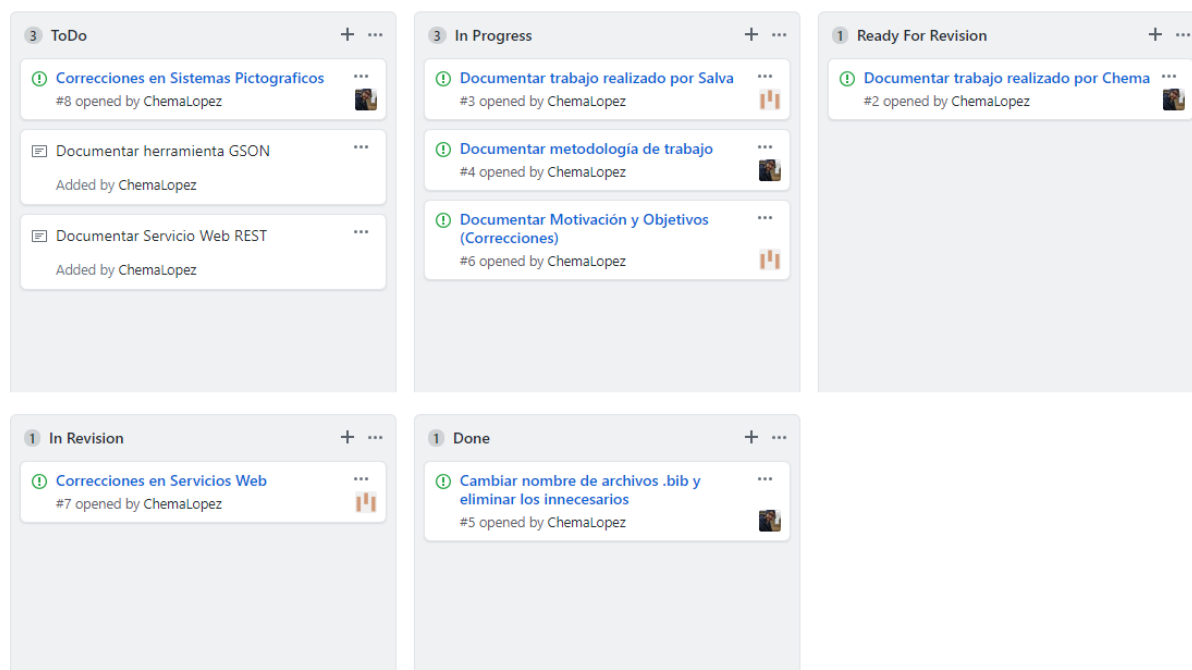


Figura 2.30: Tablero agile del proyecto

El trabajo se organiza de reunión en reunión, haciendo una estimación del tiempo que se tarda en cumplir todas aquellas tareas que han surgido en la reunión como si de Sprint de Scrum se tratase<sup>16</sup>.

### 2.6.2. Control de versiones: GitHub

Dada la naturaleza cambiante del proyecto, su duración entre otros factores es necesario además de suponer una gran ayuda una modo de mantener un control sobre las diferentes versiones tanto del código como de la documentación.

El sistema elegido ha sido GitHub dado que nos permitia tener el proyecto en privado<sup>17</sup> algo necesario dado que es un proyecto de final de grado. GitHub funciona bajo git<sup>18</sup>, un software para el control de versiones creado por Linus Torvalds<sup>19</sup>.

Git sirve para mantener un registro de todos los cambios que sufre un archivo y genera un árbol donde en cada nodo se almacenan los cambios por

<sup>16</sup><https://proyectosagiles.org/ejecucion-iteracion-sprint/>

<sup>17</sup><https://help.github.com/articles/making-a-public-repository-private/>

<sup>18</sup><https://git-scm.com/>

<sup>19</sup>[https://es.wikipedia.org/wiki/Linus\\_Torvalds](https://es.wikipedia.org/wiki/Linus_Torvalds)

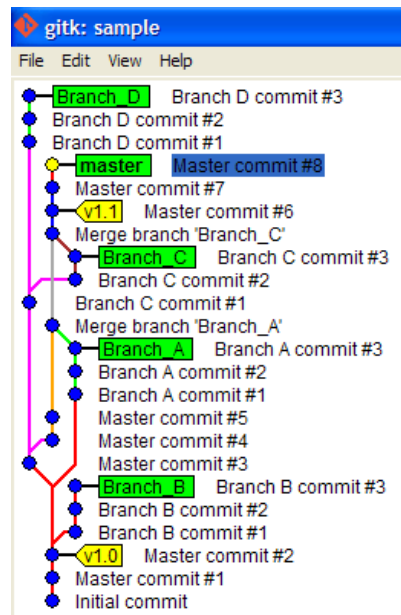


Figura 2.31: Ejemplo de árbol de Git

archivo y las diferentes ramas<sup>20</sup> en la Figura 2.31, podemos ver un ejemplo de árbol de Git. La implementación que hace GitHub de Git nos permite utilizar todos los beneficios de Git, en un entorno amigable, fácil de usar en el que ya tenemos experiencia previa, además hemos decidido utilizar la versión de escritorio<sup>21</sup>. por su comodidad y facilidad de uso, tal y como podemos ver en la Figura ?? nos permite ver los cambios, hacer commit, push y pull de una manera rápida y sencilla<sup>22</sup>.

<sup>20</sup><https://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<sup>21</sup><https://desktop.github.com/>

<sup>22</sup><https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>

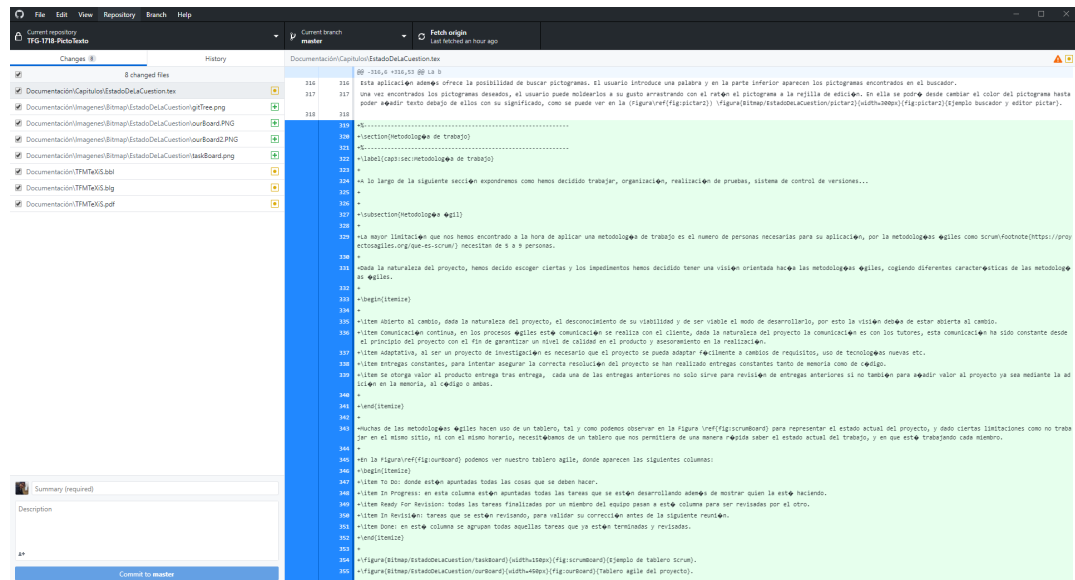


Figura 2.32: Aplicación GitHub de escritorio



# Capítulo 3

## Herramientas

A lo largo de este capítulo analizaremos las diferentes herramientas utilizadas para el desarrollo de este TFG.

### 3.1. Django

#### 3.1.1. ¿Qué es Django?

Es un framework de desarrollo web<sup>1</sup> de código abierto escrito en Python, respetando el patrón de diseño conocido como Modelo-Vista-Controlador (M-V-C). La meta principal de esta aplicación es facilitar la creación de sitios web. En Django es muy importante el re-uso de código, la conectividad y el desarrollo rápido.

1. Django da soporte de base de datos permitiendo crear los modelos de datos necesarios y a través de su API administrar y gestionar dicha base de datos. Además concede al usuario la posibilidad de poder ejecutar sus propias consultas SQL.
2. En la parte de servicios Web, Django incluye un servidor ligero que ofrece la posibilidad de realizar pruebas y trabajar en una etapa de desarrollo. Para una etapa más de producción sería más conveniente contar con otra aplicación como puede ser Apache.

### 3.2. GSON

GSON es una librería open-source la cuál nos permite convertir nuestros objetos Java en JSON o viceversa, esta librería es muy importante ya que

---

<sup>1</sup><https://tutorial.djangogirls.org/es/django/>

para lenguaje en Java no existen archivos JSON. Al haber encapsulado el Servicio web en un formato REST, esto obliga a que nuestro Servicio tenga que recibir obligatoriamente un JSON.

En este momento es donde se utiliza la librería GSON la cuál es un framework que contiene en su interior el método "from JSON", este método es vital para la construcción de nuestro Servicio web, únicamente necesita recibir el dato mandado por el usuario y el tipo de clase al que hace referencia.

Una vez obtenidos los datos mencionados, la propia librería se encarga de realizar el resto de la conversión para que el Servicio web funcione correctamente.



# Capítulo 4

## Servicios Web

A lo largo de este capítulo trataremos de explicar los diferentes servicios web, que hemos implementado, hablando sobre tecnologías y funciones que desarrollan dentro de nuestro sistema.

### 4.1. Servicio de Generación de Lenguaje Natural

Como ya hemos hablado anteriormente, hemos utilizado el framework SimpleNLG-ES<sup>1</sup>, para la fase de generación del lenguaje. Este framework está escrito en Java, lo que impide su utilización dentro de nuestro servicio web central desarrollado en Python 3<sup>2</sup>.

Este Servicio Web utiliza Apache TomCat<sup>3</sup> como contenedor, para encapsular el framework SimpleNLG-ES, y poder ser accesible desde el servicio web principal o cualquier otro servicio web del proyecto IDiLyCo<sup>4</sup>.

Este servicio web recibe un JSON con los datos necesarios para formar una frase, el cual gracias a GSON visto en ?? podemos obtener una clase Java con la cual formaremos la frase en lenguaje natural, la cual devolveremos en la respuesta del servicio.

---

<sup>1</sup><https://github.com/citiususc/SimpleNLG-ES>

<sup>2</sup><https://docs.python.org/3/>

<sup>3</sup><http://tomcat.apache.org/>

<sup>4</sup><http://nil.fdi.ucm.es/index.php?q=projects/idilyco>



# Capítulo 5

## Trabajo Individual

A lo largo de este capítulo hablaremos del trabajo realizado por cada uno de los miembros del equipo, siguiendo la metodología de trabajo explicado anteriormente en ??

### 5.1. Trabajo Individual

Como ya hemos hablado anteriormente, hemos utilizado el framework SimpleNLG-ES<sup>1</sup>, para la fase de generación del lenguaje. Este framework está escrito en Java, lo que impide su utilización dentro de nuestro servicio web central desarrollado en Python 3<sup>2</sup>.

Este Servicio Web utiliza Apache TomCat<sup>3</sup> como contenedor, para encapsular el framework SimpleNLG-ES, y poder ser accesible desde el servicio web principal o cualquier otro servicio web del proyecto IDiLyCo<sup>4</sup>.

Este servicio web recibe un JSON con los datos necesarios para formar una frase, el cual gracias a GSON visto en ?? podemos obtener una clase Java con la cual formaremos la frase en lenguaje natural, la cual devolveremos en la respuesta del servicio.

---

<sup>1</sup><https://github.com/citiususc/SimpleNLG-ES>

<sup>2</sup><https://docs.python.org/3/>

<sup>3</sup><http://tomcat.apache.org/>

<sup>4</sup><http://nil.fdi.ucm.es/index.php?q=projects/idilyco>



# Capítulo 6

## Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.



# Chapter 6

## Conclusions and Future Work

Conclusions and future lines of work.





Apéndice	<b>A</b>
----------	----------

Título

Contenido del apéndice



Apéndice	<b>B</b>
----------	----------

Título



# Bibliografía

*Y así, del mucho leer y del poco dormir,  
se le secó el cerebro de manera que vino  
a perder el juicio.*

Miguel de Cervantes Saavedra

ABRIL ABADÍN, V. C. A., I. DELGADO SANTOS C. *Comunicación Aumentativa y Alternativa*. CEAPAT, 3º edición, 2010.

ÁLVAREZ, L. L. *Los sistemas Alternativos y/o Aumentativos de Comunicación: la Comunicación Bimodal como recurso en el aula de Audición y Lenguaje*. Tfg, Universidad de Valladolid, 2013.

BASIL, C., ALMIRALL, C. y DE LA BELLACASA, R. *Comunicación aumentativa: curso sobre sistemas y ayudas técnicas de comunicación no vocal*. Colección Rehabilitación. Instituto Nacional de Servicios Sociales, 1990. ISBN 9788486852566.

BERTOLA LÓPEZ, E. *Análisis empírico de las características formales de los símbolos pictográficos Arasaac*. Tesis Doctoral, 2018-02-15.

DALE, R., MOISL, H. y SOMERS, H. *Handbook of Natural Language Processing*. Taylor & Francis, 2000. ISBN 9780824790004.

GARCÍA IBÁÑEZ, P. G., RAQUEL HERVÁS BALLESTEROS. Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural. 2004.

HEHNER, B. *Símbolos Bliss. Diccionario guía*. Subdirección General de Educación Especial, 1985.

MARÍN, F. y PÉREZ, F. *Tecnologías de ayuda en personas con trastornos de comunicación*. Colección Logopedia e Intervención Series. Nau Llibres, 2003. ISBN 9788476426821.

- MARTÍN GUERRERO, A. *PICTAR: una herramienta de elaboración de contenido para personas con TEA basada en la traducción de texto a pictogramas*. Phd, Universidad Complutense de Madrid, 2018.
- MCDONALD, E. *Sistema Bliss. Enseñanza y uso: enseñanza y uso*. Ministerio de Educación y Ciencia, Subdirección General de Educación Especial, 1985. ISBN 9788436911763.
- ROMSKI, M. A. y SEVCIK, R. A. Augmentative and alternative communication for children with developmental disabilities. *Mental Retardation and Developmental Disabilities Research Reviews*, vol. 3(4), páginas 363–368, ????
- VICENTE, M., BARROS, C., PEREGRINO, F. S., AGULLÁ, F. y LLORET, E. La generación de lenguaje natural: Análisis del estado actual. *Computación y Sistemas*, vol. 19, páginas 721 – 756, 2015. ISSN 1405-5546.
- WARRICK, A. *Comunicación sin habla*. CEAPAT, 3º edición, 2010. ISBN 0-9684186-0-0.

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –  
Bien podrán los encantadores quitarme la ventura,  
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero  
Don Quijote de la Mancha  
Miguel de Cervantes*

*–Buena está – dijo Sancho –; fírmela vuestra merced.  
–No es menester firmarla – dijo Don Quijote–,  
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero  
Don Quijote de la Mancha  
Miguel de Cervantes*

