

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



# **Ortoescritura 2.0, un entrenador ortográfico que trabaja con cualquier palabra de la Lengua Española**

---

Trabajo de diploma para optar por el título de Ingeniero en Ciencias  
Informáticas

**Autor**

**Armando Benavides Esteva**

**Tutor**

**MsC. César Labañino Rizzo**

**Pueden existir personas con mayor talento que tú,  
pero es inaceptable que exista alguien que trabaje  
más duro.**

**Derek Jeter**

A handwritten signature in black ink, appearing to read 'Derek Jeter', with a stylized, cursive script.

# Declaración de autoría

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas; así como a dicho centro, para que hagan el uso que estimen pertinente con este trabajo.

Y para que así conste firmo la presente, a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_

---

Firma del Autor

Armando Benavides Esteva

---

Firma del Tutor

César Labañino Rizzo

# Agradecimientos

A mi increíble mamá, por todo el amor, cariño, ternura... que me ha regalado en todos los años de mi vida.

A mi excelente papá por consentirme incondicionalmente.

A mi querido hermano pequeño. Espero cosas grandes de ti.

A mi abuela Queta, la quiero con todo mi corazón.

A mi abuelo Rolando por malcriarme y quererme tanto.

A abuela María por tenerme presente siempre.

A mis grandes amigos, los mejores amigos del mundo: Daniel, David, Edgar, Luisito, Michel, Yunier. Con quienes he compartido grandiosos momentos.

Al "Big Luck".

A Pablo, mi nuevo gran amigo.

A mi tutor César, por "lanzar todo el juego" conmigo. ¡Fuiste grande! A Miriam, por aguantarme todos los días.

A mi tía Isabel.

Al resto de mi familia.

A quienes me ayudaron a realizar este trabajo sin pensarlo dos veces: Yaneisy, Pavel.

A mi club en estos años de mi carrera: Nadia, Pablo, Eduardo.

A mis compañeros de equipo, los voy a extrañar.

A las muchachitas con que siempre estudiaba, me ayudaron un mundo.

A los viejos y nuevos compañeros del apartamento.

A mis compañeros de aula, un fuerte abrazo.

# Dedicatoria

A mi querida mamá, eres lo mejor que existe. A mi hermano, mi papá y mis abuelos. A Lucky. A mis mejores amigos.

## Resumen

La mala ortografía constituye un problema preocupante en las nuevas generaciones de nuestro país. El sistema educativo cubano realiza ingentes esfuerzos por erradicar este mal y el empleo de la tecnología forma parte de las estrategias factibles previstas para este fin.

El presente trabajo de investigación, tiene como objetivo aplicar un conjunto de mejoras al entrenador ortográfico Ortoescritura 1.0, concebido para entrenar competidores del concurso Hispanoamericano de Ortografía, para lograr expandir el universo de sus usuarios potenciales y a su vez disminuir el costo de la producción de sus contenidos de entrenamiento. Para el logro de estas metas, se realizó una sistematización de conceptos y enfoques metodológicos afines, se empleó el lenguaje *Transcript* y el SDK SAPI 5.0 como interfaz de programación para la tecnología TTS (Síntesis de voz), las herramientas *LiveCode* para la implementación y Visual Paradigm para la modelación con UML. La planificación y control del proceso de desarrollo de *software* se efectuó bajo la metodología RAD (*Rapid Application Development*). Se obtuvo una nueva versión del entrenador en calidad de sistema abierto, capaz de manejar cualquier palabra de la Lengua Española y se logró reducir significativamente el costo intelectual e informático del proceso de producción de sus contenidos, permitiendo la intervención de múltiples autores y de esta manera, extender el universo de sus usuarios potenciales.

**Palabras claves:** entrenador ortográfico, sistema abierto, síntesis de voz.

## Índice

Resumen.....	V
Introducción .....	1
Capítulo 1. Fundamentación teórica.....	6
Introducción .....	6
1.1 Software educativo .....	6
1.1.1 Tipos de Software Educativo .....	7
1.1.2 Software educativo abierto.....	8
1.2 Análisis de sistemas similares .....	8
1.2.1 Ortosist .....	9
1.2.2 Acentúa y Aprende .....	10
1.2.3 Jugando con las palabras.....	10
1.2.4 El fabuloso mundo de las palabras.....	10
1.2.5 El arte de las letras .....	11
1.2.6 Entrenador de Ortografía I (Aplicación mexicana) .....	11
1.2.7 Entrenador de Ortografía II (segunda edición de la serie “Ortografía para todos”) .....	11
1.3 Entrenador ortográfico Ortoescritura.....	12
1.4 Valoración del modelo didáctico del entrenador .....	15
1.5 Ambiente de desarrollo .....	16
1.5.1 Herramienta de desarrollo .....	16
1.5.2 Herramienta de modelado .....	22
1.5.3 Metodología de desarrollo .....	22
1.5.4 Tecnología de síntesis de voz .....	25
Conclusiones parciales.....	28

Capítulo 2. Presentación de la propuesta de solución .....	30
Introducción .....	30
2.1 Inicialización.....	30
2.1.1 Propuesta de sistema.....	31
2.2 Enmarcado funcional.....	31
2.2.1 Descripción de las funcionalidades.....	31
2.2.2 Jerarquía de requisitos del futuro sistema.....	34
2.3 Casos de uso.....	35
2.3.1 Patrones de casos de uso.....	35
2.3.2. Diagrama de casos de uso.....	36
2.3.3 Descripción de casos de uso.....	36
2.4 Diseño de la aplicación .....	40
2.4.1 Modelo Conceptual de Comunicación.....	40
2.4.2 Modelo Conceptual de Datos.....	41
2.5 Prototipo inicial .....	42
Conclusiones parciales.....	43
Capítulo 3 Implementación y pruebas.....	44
Introducción .....	44
3.1 Construcción.....	44
3.2 Pruebas de software.....	45
3.2.1 Métodos de prueba.....	46
3.2.2 Descripción de los Casos de Prueba.....	47
3.2.3 Resultados obtenidos.....	49
3.3 Finalización .....	50



Conclusiones parciales .....	51
Conclusiones generales .....	52
Recomendaciones .....	53
Referencias bibliográficas .....	54
Bibliografía consultada.....	55
Glosario de términos .....	56
Anexos.....	58
Anexo 1 Encuesta.....	58
Anexo 2 Descripciones de Casos de Uso.....	60
Anexo 3 Modelos Conceptuales de Comunicación (MCC).....	98
Anexo 4 Descripción de casos de prueba .....	100

## Índice de figuras

Figura 1 Tarjeta de compilación de aplicaciones para diferentes sistemas operativos.....	19
Figura 2 Fases del modelo de Desarrollo Rápido de Aplicaciones .....	25
Figura 3 Diagrama del problema del cliente.....	30
Figura 4 Diagrama de casos de uso de la aplicación. ....	36
Figura 5 Interfaz del CU Configurar modalidades de entrenamiento. ....	40
Figura 6 Modelo conceptual de comunicación del CU Realizar análisis de palabra. ....	41
Figura 7 Estructura de las bases de datos referente a los contenidos .....	42
Figura 8 Representación de las interfaces del prototipo inicial .....	43
Figura 9 Seguimiento de requisitos de la aplicación.....	45
Figura 10 Seguimiento de tratamiento de anomalías de la etapa 3 .....	50
Figura 11 Estructura de los Stacks de la aplicación final .....	51
Figura 12 Arquitectura del funcionamiento del sistema.....	51

## Índice de tablas

Tabla 1 Estructura de la base de datos del Ejercitador Ortográfico.....	12
Tabla 2 Comparación entre <i>ToolBook</i> y <i>LiveCode</i> .....	21
Tabla 3 Descripción del CU Configurar modalidades de entrenamientos.....	36
Tabla 4 Sección Seleccionar ejercicios de la modalidad Significado-Palabra.....	39
Tabla 5 Lista de palabras tipificadas. ....	47
Tabla 6 Registrar usuario en el sistema.....	47
Tabla 7 Variables del CU Registrar usuario en el sistema.....	49

## Introducción

La ortografía constituye un problema preocupante en las nuevas generaciones en Cuba. El sistema educativo cubano ha tomado medidas y planteado pautas de orden ortográfico, que influyen en las evaluaciones de los estudiantes de todos los niveles de enseñanza y hasta en la acreditación de culminación de determinados estudios. No obstante, se continúan generando quejas desde diferentes ámbitos de la sociedad como padres, maestros, profesores, entre otros, que se alarman por corroborar cada año, cómo se mantienen y acrecientan los problemas ortográficos en los jóvenes.

Un factor que históricamente ha ayudado a la correcta escritura, como la lectura de textos, está cada día menos presente en la cotidianidad de las jóvenes generaciones, lo cual hoy día se agrava con la irrupción de un audiovisualismo omnipresente y la creciente afectación que están causando la escrituras desde medios digitales, como las sesiones de chat y el uso de los dispositivos móviles.

Las características de la problemática citada, forman parte de lo que se conoce en Pedagogía con el nombre de habilidades (saber hacer) y hábitos (saber hacer automatizado), donde la repetición como mecanismo de aprendizaje juega un papel primordial, por lo que un sistema informático adecuado puede contribuir significativamente con dicho problema social.

Por otro lado, la didáctica de enseñanza de la Ortografía ofrece como uno de los métodos eficaces el denominado “Viso-audio-gnósico-motor” (observar – escuchar – comprender – escribir) cuyas características resultan factibles para ser instrumentadas desde una computadora hoy día.

Desde el año 2009 el Ministerio de Educación (MINED) desarrolló un software educativo de tipo entrenador que tiene como intención preparar a los estudiantes de preuniversitario para el Concurso Hispanoamericano de Ortografía que auspicia la “Fundación Hispanoamérica Bien Escrita”, asociación que dirige la ejecución de estas competencias.

La efectividad de los métodos didácticos que emplea el entrenador citado, se ha constatado en el contexto de este concurso. Esto se corrobora a partir del análisis de los resultados que se obtienen en el proceso de entrenamiento, lo cual se hace patente al revisar las posiciones que Cuba ocupa en este certamen, a partir de la introducción de **Ortoescritura** en el sistema de entrenamiento de los concursantes. Sin embargo, su léxico particular, constituye su mayor limitación para generalizar su uso con fines más ambiciosos, además, la implementación de sus contenidos poseen un alto costo en materia de esfuerzo intelectual (ver estructura de los datos, Tabla 1). Otro factor a tener en cuenta lo constituye el soporte en materia de ficheros de audio para el apoyo fonético que necesita el proceso de entrenamiento.

Por otro lado, desde perspectivas metodológicas de la enseñanza de la Ortografía se plantea que, aunque el concurso se restringe a una competencia a nivel de palabras, la adquisición de una buena ortografía tiene que trascender del nivel de palabra al nivel de oración o frase y que su desarrollo tiene que estar asociado a mecanismos inferenciales que se automaticen a partir del dominio de las reglas ortográficas.

En este contexto, surge la idea de aprovechar la probada eficacia de la metodología que promueve Ortoescritura y mejorar el entrenador en lo concerniente a los contextos en que pueda ser empleado.

Si se tiene en cuenta que los usuarios potenciales de un software educativo son el conjunto de individuos cuyos intereses formativos están en correspondencia con los componentes del proceso de enseñanza aprendizaje que se modelan en el software, los contenidos que maneja el software constituyen un factor determinante en la delimitación de sus usuarios potenciales. En tal sentido, si se desea ampliar el universo de usuarios potenciales de Ortoescritura, que como se ha planteado, posee una estrategia de entrenamiento cuya eficacia ha sido probada, se hace necesario diversificar los contenidos que propone el entrenador.

Diversificar los contenidos se lograría en grado superlativo si el entrenador pudiera trabajar con cualquier palabra de la Lengua Española, lo que desde la perspectiva de la informática educativa significaría transformar el software existente en un “software educativo abierto”, o sea, un software educativo con bases de datos externas e intercambiables, las cuales a su vez puedan ser producidas por expertos o maestros, no pertenecientes al equipo de desarrollo del software.

La instrumentación de estos datos, además de los inconvenientes planteados, presenta un elevado costo intelectual debido a la información que debe ser suministrada de cada palabra para lograr el proceso de entrenamiento (ver Tabla 1). Por lo antes expuesto, se hace necesario implementar un conjunto de algoritmos que permitan normalizar los datos y que con ello se simplifique la producción de los contenidos que emplea el programa para disminuir su elevado costo intelectual. Otro aspecto a tener en cuenta lo constituye el apoyo fonético que resulta imprescindible en el proceso de entrenamiento, el cual, en la versión precedente se sustenta en la existencia de un conjunto de 4130 ficheros en formato *.mp3*, uno para cada palabra objeto de entrenamiento, y los correspondientes a las oraciones y frases, lo que constituye un elevado costo de implementación.

Es necesario destacar que el software objeto de mejoramiento fue elaborado sin la aplicación de ninguna metodología de desarrollo, ni existe su código fuente, por lo que su mejoramiento no podrá beneficiarse de la existencia de artefactos que simplifiquen este proceso.

La problemática en que se desarrolla esta investigación se puede resumir de la siguiente forma:

- 1- Existencia de un software educativo tipo entrenador ortográfico cuya eficacia didáctica es admitida, sin embargo maneja un léxico muy particular, lo cual impide su generalización a otros contextos diferentes del concurso para el cual fue creado.
- 2- Por la forma en que se implementó el entrenador objeto de estudio, los contenidos no pueden ser intercambiables ni modificables y esto imposibilita aprovechar la experticia de múltiples especialistas del país para diversificar los ámbitos de aplicación del entrenador.
- 3- La producción de los contenidos de entrenamiento, en el caso de que fuera posible intercambiarlos o modificarlos, representan un costo intelectual alto, debido a la cantidad y complejidad de los atributos que conforman los datos, a esto se adiciona una tediosa generación de ficheros de audio para conseguir el apoyo fonético necesario.

En este contexto surge el **problema** de la presente investigación:

¿Cómo ampliar el universo de usuarios potenciales del entrenador Ortoescritura 1.0 bajo la condición de que disminuya el costo en la producción de su contenido?

Se plantea como **objetivo general** de la investigación:

Desarrollar una versión del entrenador Ortoescritura 1.0 en calidad de software educativo abierto, con una disminución significativa en el costo de la producción de sus contenidos, el cual permita ampliar el universo de sus usuarios potenciales.

El **objeto de estudio** de la investigación lo constituye el proceso de desarrollo de software educativo, enmarcándose en los entrenadores ortográficos asistidos por computadoras como **campo de acción**.

Teniendo en cuenta la problemática planteada y el objetivo trazado, se define como **idea a defender**: La conversión de Ortoescritura en un software educativo abierto, con disminución del costo en la producción de sus contenidos, contribuirá a ampliar el universo de sus usuarios potenciales.

## **Objetivos específicos:**

- Elaborar el marco teórico de la investigación a partir de la sistematización de los conceptos relacionados con el proceso de desarrollo de software educativo, en particular los entrenadores ortográficos.
- Concebir y diseñar una solución para mejorar el entrenador Ortoescritura 1.0 en lo concerniente a la expansión del universo de sus usuarios potenciales y la disminución del costo de la producción de sus contenidos.

- Implementar la nueva versión del entrenador.
- Validar la solución propuesta mediante pruebas de software.

## **Tareas a cumplir:**

- Estudio del entrenador Ortoescritura 1.0 y sistemas análogos e Indagación sobre regularidades lingüísticas vinculadas con procesos ortográficos.
- Evaluación de la herramienta y el lenguaje propuestos por el cliente, las tecnologías de síntesis de voz factibles y la metodología de desarrollo afin con el problema a resolver.
- Realización del levantamiento de requisitos funcionales y no funcionales de la aplicación objeto de mejoramiento.
- Modelación y elaboración de diagramas de los procesos que intervienen en la solución del problema.
- Implementación de los algoritmos que conlleven a la reducción del costo de producción de los contenidos (estructura de datos y eliminación de los ficheros de audio).
- Implementación del resto de las funcionalidades no asociadas con el problema.
- Realizar las pruebas necesarias para asegurar la calidad de la solución propuesta.

Para llevar a cabo la investigación se utilizaron los siguientes **métodos teóricos**:

**Histórico – Lógico:** con la intención de valorar el desarrollo y evolución diacrónica de programas con características análogas a Ortoescritura.

**Analítico-Sintético:** para identificar singularidades de los productos estudiados y obtener sus características fundamentales.

**Inductivo-deductivo:** para indagar en los procesos lingüísticos que intervienen en el problema y la generalización de algoritmos asociados con las reglas ortográficas.

**Análisis documental:** para consultar literaturas especializadas sobre la materia.

**Modelación:** para la creación y esbozo de los modelos y prototipos en los procesos de diseño e implementación.

Se hizo uso del **método empírico**:

**Encuesta:** con el objetivo de recopilar información, criterios y opiniones sobre las insatisfacciones de usuarios del entrenador Ortoescritura 1.0.

## **Estructura Capítular**

### **Capítulo 1: Fundamentación Teórica**

Se describe el marco teórico de la investigación realizada, así como la problemática en que ésta se enmarca. Se exponen los conceptos vinculados con el dominio del problema y se indagan y analizan los componentes relacionados con el problema desde productos que poseen objetivos similares. Se valora la herramienta, el lenguaje, tecnologías afines y la metodología que se empleará en la solución del problema.

### **Capítulo 2: Presentación de la propuesta de solución**

En este capítulo se propone una solución al problema de investigación. Se definen los requerimientos funcionales y no funcionales del sistema, y se generan los artefactos y modelos necesarios según la metodología adoptada, como el diagrama de casos de uso, modelo conceptual de comunicación, modelo conceptual de datos y el prototipo inicial de la aplicación.

### **Capítulo 3: Implementación y pruebas**

En este capítulo se refina el prototipo mediante iteraciones sucesivas hasta llegar al logro de la aplicación. Se documentan los tipos de pruebas a realizar y se exponen los resultados obtenidos.



# Capítulo 1. Fundamentación teórica

## Capítulo 1. Fundamentación teórica

### Introducción

Como se observa el problema a resolver tiene como objeto de estudio las herramientas educativas, y en particular las de tipo Entrenador, vinculado con la ortografía de la lengua española. Como se ha evidenciado, la solución del problema guarda estrecha relación con los contenidos que maneja el sistema; se plantea como necesidad, ampliar significativamente el universo de usuarios potenciales haciendo posible la diversificación de sus contenidos y este proceso debe lograrse disminuyendo los costos en la producción de los contenidos (textuales y audibles) objeto de entrenamiento. De este análisis se desprende el carácter interdisciplinario que posee el dominio del problema, en el que como se infiere, intervienen tanto conceptos de la informática, la informática educativa, como los derivados de ámbitos de carácter lingüístico.

### 1.1 Software educativo

En diferentes fuentes consultadas se define el “software educativo” de diferentes maneras, las que a priori, parecerían ser equivalentes.

Jaime Sánchez (1), define el concepto genérico de Software Educativo como, “...cualquier programa computacional cuyas características **estructurales y funcionales** sirvan de apoyo al proceso de enseñar, aprender y administrar”.

Pere Marques (2) plantea que: “Se considera un software educativo, a todo aquel programa para ordenador, creado con la **finalidad específica** de ser utilizado como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje”.

La primera definición posee un carácter funcional, mientras que la segunda posee un carácter intencional. El autor se afilia a la segunda definición por considerarla más precisa. A partir de la definición de Sánchez (1), aplicaciones como Word o Excel podrían ser considerados como software educativos, o sea, se plantea que si el software puede ser entendido como medio didáctico, entonces puede ser considerado un software educativo y en la opinión del autor esta definición puede acarrear conflictos entre la definición de parte/todo, ya que en principios, cualquier software, por singular que este sea, en un momento dado, es objeto de aprendizaje y en esas circunstancias sería un medio didáctico, de lo que se derivaría, que en esas circunstancias sería un software educativo, o sea que de lo antes dicho, prácticamente todos los software serían, según la definición de (1), software educativos.

# Capítulo 1. Fundamentación teórica

Estos se caracterizan por ser fáciles de usar, tener gran interactividad con el usuario, tener una finalidad didáctica, apoyarse en tecnología multimedia y poder adaptarse a la capacidad de aprendizaje y ritmo de trabajo de los usuarios. En su estructura intervienen generalmente una interfaz destinada a interactuar con el usuario, bases de datos para almacenar la información y algoritmos en función de las acciones del programa. (2)

Se concluye que este planteamiento posee elementos poco rigurosos y con relativa ambigüedad. La idea referente a la facilidad de su uso (usabilidad), es ambiguo en la definición, ya que desestima el concepto de curva de aprendizaje en el contexto de los software educativos y la interactividad es concebida en un grado superlativo, cuando es conocido que esta característica depende del tipo de software educativo de que se trate.

El autor (2) expresa que este tipo de software suele tener diferentes funciones:

- **Función Informativa:** El contenido se presenta con información estructurada de acuerdo a la actividad que vayan a realizar.
- **Función Instructiva:** Se enseña y orienta a los usuarios sobre la actividad a realizar. Promueve determinadas acciones para facilitar el logro del objetivo.
- **Función Motivadora:** Se crea una atracción y conexión hacia el software con el objetivo de adquirir las habilidades previstas.
- **Función Investigadora:** Se ofrecen determinados entornos donde se puedan buscar determinadas informaciones.
- **Función Evaluadora:** Debido a la interactividad propia de estos programas, se puede evaluar el trabajo que se va realizando con los usuarios.

El autor considera que debería añadirse la función educativa, vinculada con la formación en valores, que pueden tener el software educativo, en tanto que constituyen medios de enseñanza, o sea función formativa.

## 1.1.1 Tipos de Software Educativo

Pere Marqués plantea que los tipos de software existentes son: (2)

- **Tutoriales:** Son programas que a partir de informaciones y la realización de actividades previstas dirigen y tutorizan el trabajo de los alumnos.

# Capítulo 1. Fundamentación teórica

- Programas de ejercitación. Se limitan a proponer ejercicios auto-correctivos de refuerzo sin proporcionar explicaciones conceptuales previas. Su estructura puede ser: lineal (la secuencia en la que se presentan las actividades es única o totalmente aleatoria), ramificada (la secuencia depende de los aciertos de los usuarios) o tipo entorno (proporciona a los alumnos herramientas de búsqueda y de proceso de la información para que construyan la respuesta a las preguntas del programa).
- Bases de datos: Brindan datos organizados según determinado criterio facilitando su exploración y consulta selectiva.
- Simuladores: Presentan un modelo o entorno dinámico de fácil exploración y/o modificación que apoya el aprendizaje significativo por descubrimiento.
- Constructores: Software con un entorno programable que facilita al usuario el uso de elementos simples para construir elementos más complejos.
- Programas herramientas: Proporcionan un entorno instrumental para facilitar ciertos trabajos generales de tratamientos de información.

Como se observa, Ortoescritura se enmarca perfectamente en esta taxonomía, en la categoría de los programas que proponen ejercicios auto-correctivos.

## 1.1.2 Software educativo abierto

En la nomenclatura de la Informática Educativa se considera un sistema abierto (programa educativo abierto) al software educativo cuyos datos se almacenan en bases de datos externas, intercambiables o modificables, lo cual posibilita diversificar los contenidos que se le presentan al usuario sin necesidad de alterar el código de la aplicación. Generalmente, los software educativos abiertos ofrecen mecanismos o interfaces ergonómicas para la gestión de los datos, con lo que el software adquiere ciertos atributos de un proceso que caracteriza a las herramientas de autor. (3)

Esta definición resulta de vital importancia en el contexto de esta investigación, se destaca la existencia de bases de datos externas, intercambiables y modificables.

## 1.2 Análisis de sistemas similares

El Ministerio de Educación dispone de más 200 **software educativos** concebidos bajo el concepto de hiperentornos de aprendizaje que no son más que **ambientes virtuales de aprendizaje**, diseñados

# Capítulo 1. Fundamentación teórica

sobre la base de diversas tipologías de software educativo, sustentado básicamente en tecnología hipertexto. Entre los que se listan: (4)

- “A Jugar” con 12 productos para las enseñanzas Preescolar y Especial.
- “Multisaber” con 32 software para la enseñanza Primaria y Especial.
- “El Navegante” con 10 plataformas para la enseñanza Secundaria Básica.
- “Futuro” con 16 hiperentornos para las enseñanzas Preuniversitario, Enseñanza Técnica Profesional y Adulto.
- “Premédicos” con 13 software para la formación de médicos latinoamericanos.
- “Historia Patria” con 13 productos.

Adicionalmente se cuenta con productos monotemáticos, no pertenecientes a colecciones y clasificados en tipologías diversas, como es el caso de Ortoescritura, el entrenador ortográfico objeto de esta tesis.

Del amplio espectro de software educativos mencionados, se valoró aquellos títulos cuyo contenido o tipología pudiera estar en correspondencia con el campo de esta investigación. El estudio realizado se centró en las características que intervienen en el problema de la presente investigación:

- Público al que va dirigido.
- Tratamiento de sus datos (sistema abierto o cerrado).
- Costo intelectual de la producción de sus contenidos.
- Recursos de apoyo que se emplea para la factura fonética.

## 1.2.1 Ortosist

**Público:** Estudiantes de secundaria básica.

**Contenidos:** Uso de grafemas (s, c, z, x / b, v / ll, y / g, j / h)

**Tipología:** Entrenador

Este software presenta ejercicios de manera secuencial y aleatoria. Constituye un software educativo cerrado. Se deduce que el costo intelectual de la producción de sus retroalimentaciones es alto, debido a la precisión y puntualidad en que ésta se manifiesta. No emplea facturas fonéticas.

# Capítulo 1. Fundamentación teórica

## 1.2.2 Acentúa y Aprende

**Público:** Estudiantes de Primaria segundo ciclo.

**Contenidos:** Reglas ortográficas. (RAE), - Ejercicios de acentuación con: agudas, llanas, esdrújulas, sobreesdrújulas, hiatos, diptongos, triptongos, monosílabos, acento diacrítico, palabras con enclíticos, sinónimos, antónimos, parónimos, homófonos, palabras de doble y triple escritura.

**Tipología:** Hiperentorno de aprendizaje.

Este software combina sesiones tutoriales y de ejercitación. Constituye un software educativo cerrado. No es posible valorar el costo intelectual de la producción de sus contenidos. Emplea ficheros de audio no comprimido para el logro de sus facturas fonéticas.

## 1.2.3 Jugando con las palabras

**Público:** Estudiantes de Primaria 4to grado y segundo ciclo.

**Contenidos:** Gramática. Reglas ortográficas en forma de prosa y verso. - Ejercicios de acentuación con: agudas, llanas, esdrújulas, sobreesdrújulas y monosílabos. Juega con sinónimos y antónimos,

**Tipología:** Hiperentorno de aprendizaje.

Este software presenta igualmente diferentes tipologías de software. Constituye un software educativo cerrado. El costo intelectual de la producción de sus contenidos es alto, lo que se pudo constatar a partir del guion del software. Emplea ficheros de audio comprimido para el logro de sus facturas fonéticas.

## 1.2.4 El fabuloso mundo de las palabras

**Público:** Estudiantes de secundaria básica.

**Contenidos:** Gramática. Reglas ortográficas. - Ejercicios de acentuación con: agudas, llanas, esdrújulas, sobreesdrújulas, hiatos, diptongos, triptongos, monosílabos, acento diacrítico, palabras con enclíticos, sinónimos, antónimos, parónimos, homófonos, palabras de doble y triple escritura.

**Tipología:** Hiperentorno de Aprendizaje.

Al clasificarse como Hiperentorno de Aprendizaje, combina diferentes tipologías de software. Constituye un software educativo cerrado aunque instrumenta un mecanismo de carga de módulos complementarios en calidad de plug-in, mediante un tratamiento basado en registros del sistema operativo. El costo intelectual de la producción de sus contenidos es alto lo que se pudo constatar a

# Capítulo 1. Fundamentación teórica

partir del guion del software. Emplea ficheros de audio comprimido para el logro de sus facturas fonéticas y síntesis de voz de primera generación, mediante una tecnología denominada Agentes de Microsoft, para generar asistencia y retroalimentaciones al usuario.

## 1.2.5 El arte de las letras

**Público:** Estudiantes de preuniversitario.

**Contenidos:** Literatura. Gramática. Acentuación gráfica. Tipología: Hiperentorno de Aprendizaje. Combina diferentes tipologías de software. Constituye un software educativo cerrado aunque instrumenta mecanismos de carga de módulos complementarios en calidad de plug-in mediante un tratamiento basado en registros del sistema operativo. El costo intelectual de la producción de sus contenidos es alto ya que se empleó la misma metodología de su predecesor “El Fabuloso Mundo de las Palabras”. Emplea ficheros de audio comprimido para el logro de sus facturas fonéticas y síntesis de voz de primera generación, mediante una tecnología denominada Agentes de Microsoft, para generar asistencia y retroalimentaciones al usuario.

## 1.2.6 Entrenador de Ortografía I (Aplicación mexicana)

**Público:** está dirigida fundamentalmente a niños entre 6 y 11 años.

**Contenidos:** Reglas de acentuación. Es el primer número de la serie “Ortografía para todos”.

Este entrenador permite seleccionar varios tipos de ejercicios que se distribuyen aleatoriamente y se encuentran apoyados en imágenes, pronunciación de palabras e información sobre la regla de acentuación y significado en cuestión. Brinda además, la posibilidad de obtener el historial del estudiante, su calificación, así como estadísticas de sus resultados. (5)

Por ser software privativo, no se pudo constatar cómo se produce el costo de la producción de sus contenidos, ni si emplea mecanismos para apoyo fonético.

## 1.2.7 Entrenador de Ortografía II (segunda edición de la serie “Ortografía para todos”)

**Público:** jóvenes entre 12 y 16 años.

**Contenido:** Reglas de acentuación.

Permite ejercitar a partir de una cantidad de ejercicios seleccionados; como mínimo 8. Presenta el juego sopa de letras, donde se descubren las palabras con un marcado énfasis ortográfico a partir de su significado. Incluye crucigramas, que tienen como objetivo, descubrir palabras y otros elementos relacionados con palabras de difícil comprensión, ubicándolas donde correspondan. (6)

# Capítulo 1. Fundamentación teórica

Por ser software privativo, no se pudo constatar cómo se produce el costo de la producción de sus contenidos, ni si emplea mecanismos para apoyo fonético.

## 1.3 Entrenador ortográfico Ortoescritura

El entrenador ortográfico Ortoescritura es un programa desarrollado por el MINED en el 2009, bajo las bases del Concurso Hispanoamericano de Ortografía, para entrenar habilidades ortográficas asociadas exclusivamente con el léxico de la competencia. Posterior al estudio de este software, se enumeran sus características fundamentales:

- Se sustenta en la simulación del proceso de dictado, para lo cual se ha previsto un sistema de ficheros de audio en correspondencia con las palabras clasificadas en los 4 niveles de dificultad mencionados.
- El entrenamiento se presenta en 2 modalidades: palabra/significado y oración/frase.
- La presentación de los ejercicios dentro de una modalidad se lleva a cabo a partir de 2 criterios, en 2 modalidades: Una es el nivel (I, II, III y IV) y la otra es la presentación de los contenidos (secuencial / aleatoria).
- Los resultados del proceso de entrenamiento se registran en un módulo denominado “traza” que brinda información acerca del progreso de los estudiantes con el entrenador.
- La estructura de los datos para conformar los contenidos de entrenamientos de la modalidad palabra-significado es la siguiente:

Tabla 1 Estructura de la base de datos del Ejercitador Ortográfico.

Atributo	Tipo de dato	Descripción
<b>Palabra</b>	Cadena	Palabra a tratar.
<b>Significado</b>	Cadena	Significado de la palabra.
<b>PalEnSil</b>	Cadena	Palabra a tratar dividida en sílabas.
<b>PoSilTon</b>	Entero	Posición de la sílaba tónica
<b>Clasif</b>	Cadena	Clasificación de la palabra según su acentuación (aguda, llana, esdrújula o sobresdrújula).
<b>Diptongo</b>	Boolean	Si la palabra presenta diptongo o no.
<b>Hiato</b>	Boolean	Si la palabra presenta hiato o no.
<b>AdvMente</b>	Boolean	Si la palabra es un adverbio terminado en “mente” o no.
<b>Diacrítico</b>	Boolean	Si la palabra presenta acento diacrítico o no.

# Capítulo 1. Fundamentación teórica

Este entrenador ortográfico cuenta con un total de 4130 palabras colocadas en 4 niveles diferentes de dificultad, 1085 palabras en el primer nivel (elemental), 1273 en el segundo (básico), 1019 en el tercero (semiculto) y 753 en el cuarto (culto). Contiene además, 214 oraciones con sus respectivos apoyos fonéticos en ficheros .mp3.

## Proceso de entrenamiento

Existen dos modalidades, cada una de las cuales posee a su vez dos niveles de trabajo:

a) Modalidad: Palabra- significado

- Familiarización y análisis
- Dictado e inferencia

b) Modalidad: Oración- frase

- Escritura con apoyo
- Escritura libre

La presentación de los ejercicios se podrá hacer con dos variantes:

- Secuencial (permite planificar múltiples entradas planificadas) Ej. 1-20, 21-40, etc.
- Al azar (solo se indica la cantidad)

Y el programa decide los que serán presentados.

## Modalidad Palabra- significado: Familiarización y análisis

Este nivel presenta listados de palabras con el objetivo de que el estudiante interactúe con ellas, escuche su pronunciación, a la vez que pueda leer su significado y opcionalmente realice un análisis paso a paso sobre la existencia o no de tilde en ella.

## Acerca del análisis ortográfico

El programa propone un diálogo interactivo a partir de la palabra seleccionada, que permitirá recopilar una serie de datos acerca de la palabra y que finalmente permitirán inferir si la palabra tratada debe llevar tilde o no. De manera complementaria, se sugiere revisar la regla que se vincula con la palabra tratada desde el punto de vista de su acentuación gráfica. Este diálogo interactivo contempla: la división de la palabra en sílabas; la localización de la sílaba tónica; la clasificación de palabras en agudas, llanas, esdrújulas, y sobresdrújulas; la identificación de diptongos; hiatos y el tratamiento de los adverbios terminados en mente.



# Capítulo 1. Fundamentación teórica

## **Palabra- significado: Dictado e inferencia**

Este nivel presenta palabras que deberán ser escritas correctamente por los estudiantes a partir de un dictado.

Los elementos que intervienen en este ejercicio son:

- La escucha de la palabra
- La lectura del significado de la palabra
- Una pista sintáctica de la palabra generada a partir de un algoritmo basado en la probabilidad de existencia de accidentes ortográficos, sustentados en la ocurrencia de los grafemas (s, c, z, x/b, v/g, j/y, ll/h) en situaciones particulares.
- La presentación de la propia palabra de manera intermitente.

## **Oración-frase: Escritura con apoyo**

Este nivel presenta oraciones o frases que deberán ser completadas correctamente por los estudiantes a partir de su escucha.

Los elementos que intervienen en este ejercicio son:

- Oración o frase horadada según un algoritmo que procesa los posibles accidentes ortográficos (s, c, z, x/b, v/g, j/y, ll/h), uso de mayúsculas, acentuación gráfica y signos de puntuación.
- La lectura de la oración o frase retenida por espacio de un segundo por palabra.
- La escucha de la oración o frase.

## **Oración-frase: Escritura libre**

Este nivel presenta oraciones o frases que deberán ser escritas correctamente por los estudiantes a partir de su escucha.

Los elementos que intervienen en este ejercicio son:

- La lectura de la oración o frase por espacio de retención de un segundo por palabra.
- La escucha de la oración o frase

## **Acerca del análisis ortográfico sobre los errores cometidos**

Si el error cometido es de tipo acentual, el sistema propone el análisis descrito en la modalidad I, por el contrario si el error se produce sobre un grafema, uso de mayúscula o signo de puntuación, el programa propondrá consultar las reglas que en sentido general y de manera probabilística, pudieran estar

# Capítulo 1. Fundamentación teórica

relacionadas con el error cometido. Este tratamiento se considera un efecto de retroalimentación, no exacto o débil, cuya consecuencia obedece al carácter difuso o no totalmente estructurado del sistema gramatical del lenguaje natural. (7)

Traza: el programa guardará la fecha y la hora, el tipo de ejercicio que se ha realizado, el contenido con que se ha trabajado, la forma de acceso a este contenido (secuencial o al azar), la ronda o veces que se ha realizado este ejercicio en una misma sesión de trabajo y la oración o frase ejercitada, así como las palabras erradas por el estudiante, acompañadas de su patrón correcto.

Complementariamente registrará:

- Si la sesión de trabajo fue concluida o abortada.
- El total de oraciones o frases seleccionadas y el total de realizadas.
- La cantidad de puntos obtenidos (en dependencia de la cantidad de accidentes ortográficos existentes en la oración o frase).
- El tiempo consumido en la escritura de las oraciones o frases.
- Un índice de efectividad como la relación porcentual entre lo que se debía lograr y lo hecho correctamente.

## Limitaciones

Este programa, que solamente funciona en el sistema operativo Windows, solo puede utilizar el juego de palabras previstos y su impedimento adicional radica en el apoyo fonético que se necesita para su funcionamiento, el empleo de ficheros de sonido, así como el alto costo intelectual para la conformación de sus datos, aspectos que constituyen barreras para convertir el uso del programa en diversos escenarios.

Resulta importante destacar que una condición contraproducente en el contexto del problema, es la no existencia de artefactos y/o documentos resultantes de la aplicación de alguna metodología de desarrollo, ni de las fuentes del entrenador original, lo cual hubiera sido un potente catalizador en el proceso del mejoramiento del producto.

## 1.4 Valoración del modelo didáctico del entrenador

Con el objetivo de valorar el modelo didáctico del entrenador, se realizó una encuesta a un total de 52 estudiantes, incluyendo 16 competidores del concurso de la escuela vocacional *Vladimir I. Lenin*, donde el resultado de las encuestas realizadas se pueden visualizar en: Ver Anexo 1. Después de efectuada la encuesta se concluye:

# Capítulo 1. Fundamentación teórica

- Con respecto a Ortoescritura, los estudiantes se encuentran satisfechos con la preparación que brinda el *Ejercitador Ortográfico con vistas al Concurso Hispanoamericano de Ortografía*.
- Los estudiantes consideran el dictado de palabras como la herramienta más útil del entrenador.
- Existe un interés significativo por relacionarse con el vocabulario español, o sea, la utilidad de conocer la pronunciación, escritura, y definición de las palabras.
- Las palabras que el sistema incluye de forma predeterminada, no son suficientes para los estudiantes si se trata de un entrenamiento ortográfico pleno, más allá del Concurso Hispanoamericano de Ortografía.
- La extensión del vocabulario sobre asignaturas específicas, parece un punto de interés de los estudiantes.

## 1.5 Ambiente de desarrollo

A continuación se expone el fundamento de las herramientas utilizadas para el desarrollo de la aplicación.

### 1.5.1 Herramienta de desarrollo

#### *LiveCode*

El Ministerio de Educación se encuentra enfrascado en la problemática de desarrollos multiplataforma y dispositivos móviles, para lo cual ha identificado una herramienta de desarrollo denominada *LiveCode*, antiguamente *Runtime Revolution*, que hoy día es software libre bajo la licencia GPLv3. Por los motivos expuestos, el cliente la propuso como herramienta para el desarrollo de las mejoras de Ortoescritura.

*LiveCode* es una herramienta de desarrollo rápido que integra un lenguaje de programación de alto nivel (*Transcript*), editor y depurador de código, así como herramienta de diseño de interfaces de usuario. Incluye soporte multiplataforma, tiene características que le permiten administrar y consultar datos. Trabaja sobre la filosofía de su propio lema “**Escríbalo una vez, ejecútelo donde quiera**” (***Write once, run anywhere***), se programa una vez y se compila para varias plataformas incluyendo dispositivos móviles soportados en *Android* e *IOS*. (8)

Con *LiveCode* es factible desarrollar aplicaciones educativas, juegos, títulos multimedia, kioscos publicitarios online, diseños de prototipos y GUI (Interfaz Gráfica de Usuario, por sus siglas en inglés) para bases de datos, aplicaciones para Internet y CGI (Interfaz de Entrada Común, por sus siglas en inglés), así como soluciones empresariales. Las aplicaciones *LiveCode* se construyen alrededor del

# Capítulo 1. Fundamentación teórica

paradigma de “pila de tarjetas”. Las pilas son una colección de recursos y manipuladores de eventos en las que se colocan los diferentes elementos o controles.

Características: (9)

- Herramientas de desarrollo rápido de GUI: Presenta un IDE (Entorno de desarrollo Integrado, por sus siglas en inglés,) y un lenguaje de alto nivel denominado *Transcript* basado en objetos y dirigido por eventos.
- *Transcript*, como la mayoría de los lenguajes de script es débilmente tipado y sus sintaxis proviene de *Hypertalk*, lenguaje proveniente del mundo Macintosh.
- Es un lenguaje de tercera generación de tipo “*English like*” por su semejanza con el lenguaje natural.
- Auto completamiento de las principales estructuras. Diccionario con las funciones y comandos que contiene.
- Es un lenguaje extensible por diferentes vías: Plug-ins y el empleo de SDK (Kit de Desarrollo de Software, por sus siglas en inglés).
- Amplia gama de funciones para el tratamiento multimedia y el manejo de bases de datos.
- *LiveCode* posee una GUI para varias plataformas y *widgets* prefabricados con apariencia nativa para múltiples sistemas operativos. (Windows, Linux, MacOS, Android, IOS)
- Fácil de programar: Se puede desarrollar la aplicación en tiempo de ejecución. Utiliza inspector y editor de propiedades, depurador de código e inspector de mensajes con una jerarquía personalizable.
- Objetos: Contiene alrededor de 38 tipos de objetos preconstruidos. Control total desde el lenguaje para todos los objetos. Herencia entre objetos.
- Multimedia y gráficos: Soporte para animaciones, reproductor de video, efectos de transición, grabación de sonidos y ejecución de formatos de sonidos como .WAV, .AU y .AIFF. Gráficos vectoriales y soporte para formatos de imagen como .JPEG, .PNG, .GIF y .PICT.
- Tratamiento de Audio y Video: Contiene un conjunto de comandos y funciones para capturar videos desde aplicaciones y el control de la configuración de estos. Permite el uso de síntesis de voz. Control de la configuración de la voz, velocidad, tono y volumen.

# Capítulo 1. Fundamentación teórica

- Redes: Muestra de películas, imágenes y texto desde cualquier URL (Localizador Uniforme de Recursos, por sus siglas en inglés). Permite el uso de servicios HTTP (Protocolo de Transferencia de Hipertextos, por sus siglas en inglés) y FTP (Protocolo de Transferencia de Archivos, por sus siglas en inglés). Carga y descarga de ficheros desde y hacia sitios FTP. Control de dispositivos en los puertos.
- XML (Lenguajes de Marcas Extensible, por sus siglas en inglés): Posibilita la creación y manipulación de contenido XML. Análisis sintáctico, trabajo con ficheros de este tipo y bibliotecas de funciones XML-RPC (Protocolo de Llamada a Procedimiento Remoto, por sus siglas en inglés).
- Acceso a bases de datos SQL (Lenguaje de Consulta Estructurado, por sus siglas en inglés): Fácil conexión a las bases de datos. Acceso directo a bases de datos como Oracle, PostgreSQL y MySQL.
- Soporte UNICODE: Utiliza este estándar de codificación para la entrada y visualización de idiomas que no utilicen el alfabeto romano. Importa y exporta ficheros de texto con este soporte. Convierte texto UNICODE con otras codificaciones.
- Extensibilidad del lenguaje: Permite utilizar SDK, para la creación de recursos externos en cualquier lenguaje compilado. Utiliza tecnologías específicas de plataformas.
- Procesamiento de textos: Uso de expresiones en inglés para el manejo de fragmentos de textos. Vasto soporte para el estilo y formateo del texto. Soporte para la búsqueda y ordenamiento de texto. Formatos HTML y RTF (Formato de Texto Enriquecido, por sus siglas en inglés).
- Multiplataforma: La plataforma de desarrollo corre en todos los Sistemas Operativos. Escríbalo una vez, ejecútelo donde quiera. Con apariencia y comportamientos acorde a la plataforma, además, brinda vistas previas en cada plataforma. La siguiente figura muestra la interfaz de compilación de aplicaciones hacia diferentes sistemas operativos:

# Capítulo 1. Fundamentación teórica

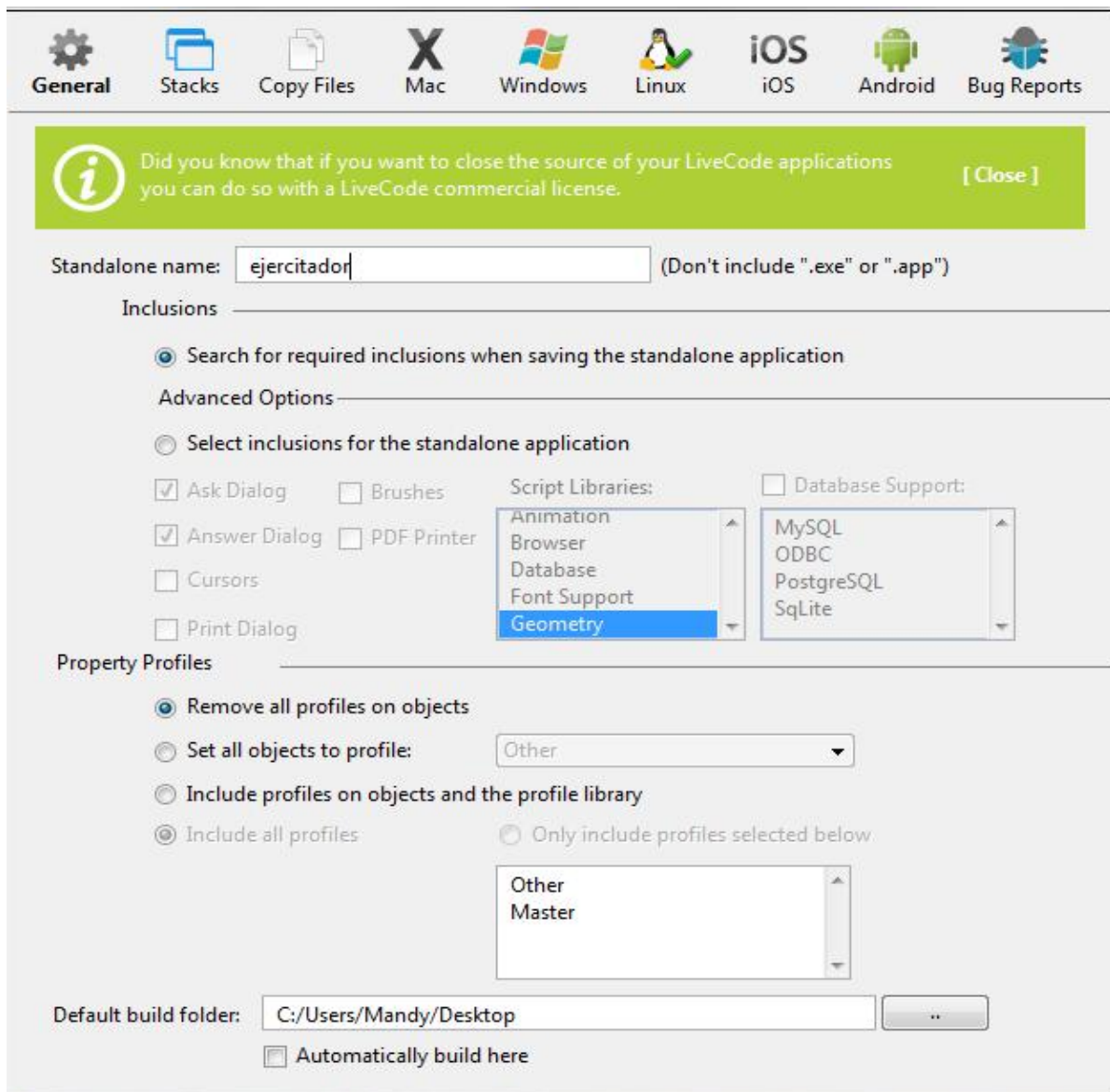


Figura 1 Tarjeta de compilación de aplicaciones para diferentes sistemas operativos.

Las características expuestas anteriormente pueden ser fácilmente consideradas como ventajas de la herramienta. *LiveCode* tiene también elementos que juegan en su contra como son (8):

- Lenguaje explícito: *Transcript* puede llegar a ser un arma de doble filo pues es tan verbal que llega a romper el paradigma que utilizan en las instrucciones los demás lenguajes. En un ejemplo de asignación como “a=b+c”, *Transcript* no admite este tipo de declaración, permitiendo en cambio las formas “put b+c into a” o “add b+c to a”; llegando a ser más engorroso a la hora de asignar nuevos valores a propiedades de objetos contenidos en otros objetos.

# Capítulo 1. Fundamentación teórica

- Escases de objetos pre construidos: No se dispone de una gran cantidad de objetos para cubrir tareas comunes de producción de títulos multimedia como botones de navegación y en el caso de software educativo, los componentes para el trabajo con cuestionarios interactivos.

## Selección de la herramienta de desarrollo

El entrenador preexistente fue desarrollado con la herramienta de autor propietaria *ToolBook*, de SumTotal (antiguamente Asymetrix), altamente difundida en el mundo de la Informática Educativa, conjuntamente con las herramientas Director, Athorware y Flash, de Macromedia (hoy pertenecientes a Adobe) y podría resultar sensato realizar las mejoras de Ortoescritura con esta herramienta para reutilizar gran parte de la funcionalidad que permanece inalterable en la nueva versión, por lo que independientemente del requerimiento planteado por el cliente en relación con *LiveCode*, se realizó el siguiente análisis: (3)

- *ToolBook* es una herramienta para resolver problemas de la informática educativa y su lenguaje de programación OpenScript es de alto nivel y extensible, pero es de carácter privativo y su costo es elevado.
- *ToolBook* desarrolla tanto aplicaciones de escritorio como aplicaciones Web, pero sus aplicaciones nativas (de escritorio) solo pueden ejecutarse en el sistema operativo Windows y aunque no constituye un requisito no funcional el que el mejoramiento de Ortoescritura sea multiplataforma, se conoce lo inminente de esta problemática en el país.
- El manejo de bases de datos de *ToolBook* de manera nativa se circunscribe exclusivamente a Access y Paradox (base de datos de Borland hoy día descontinuada)
- El carácter abierto del programa apuntaba al empleo de la tecnología de síntesis de voz y para emplear esta desde *ToolBook*, resultaba necesario acudir a la extensibilidad del lenguaje mediante enlaces dinámicos de bibliotecas (*dlls*) que habría que construir o de lo contrario usar un *ActiveX* comercial no factible para el MINED, que terminaría siendo una tecnología cerrada al mundo Microsoft.

Del análisis documental realizado, el autor concluye que tanto *LiveCode* como *ToolBook* provienen del mundo *Apple*, derivados de un antecesor denominado *HyperCard* sustentados en una familia de lenguajes provenientes del *Hypertalk* con características muy similares en lo concerniente a su sintaxis denominada "English-Like" por su cercanía al lenguaje natural. (*OpenScript* en *ToolBook* y *Transcript* en *LiveCode*)

# Capítulo 1. Fundamentación teórica

- Existe una homologación impresionante entre las metáforas de ambos sistemas, en materia del espectro de objetos preconstruidos (“*building objects*”) como se muestra en la siguiente tabla:

Tabla 2 Comparación entre *ToolBook* y *LiveCode*.

En <i>ToolBook</i>	En <i>LiveCode</i>	Algunas características
<b>Book</b>	<b>Stack</b>	Estructura básica de la aplicación
<b>Page</b>	<b>Card</b>	Pantalla de información
<b>BackGround</b>	<b>BackGround</b>	Objeto común responsable de contener objetos comunes a varias instancias del objeto Page/Card
<b>Viewer</b>	<b>Windows</b>	Tratamiento de ventanas, cuadros de diálogos, paletas, etc. Como page/cards instanciados en un visor especial.
<b>Fields, buttons, line, polygon, list-boxes, angledline, arc, curve, rectangle, ellipse, irregularpolygon, comboBox</b>	<b>Fields, buttons, line, polygon, list-boxes, angledline, arc, curve, rectangle, oval, frehandpolygon, comboBox</b>	Objetos (en el sentido informático) preexistentes en la paleta de ambas herramientas.
<b>Openscript</b>	<b>Transcript</b>	Derivados ambos de <b>Hypertalk</b> con grandes similitudes sintácticas y semánticas.
<b>IDE de ToolBook</b>	<b>IDE de LiveCode</b>	Muy similares

Después de analizar las características de las anteriores herramientas el autor corrobora la pertinencia de *LiveCode* para desarrollar la aplicación. *LiveCode* se presenta como una herramienta adecuada debido, entre otros factores ya mencionados, a su similitud con la herramienta precedente en 3 aspectos fundamentales: metáfora, IDE y lenguaje de programación, además de que *LiveCode* supera con creces algunas de las limitaciones presentes en *ToolBook* como son lo referente al manejo de bases de datos y el hecho de que de manera nativa *Transcript* posee funciones predefinidas para el manejo de la síntesis de voz (*revSetSpeechSpeed*, *revUnloadSpeech*, *revSetSpeechVolume*, *revIsSpeaking*, *revSpeechVoices*), aspecto vital para el logro del carácter “abierto” de la aplicación. Como valor añadido, la característica “**Write once, run anywhere**” constituye un argumento importante para futuro desarrollo del entrenador en nuevas versiones.



# Capítulo 1. Fundamentación teórica

Como colofón, la documentación de *LiveCode* se afilia con una metodología concreta de desarrollo de Software, la metodología RAD, que resulta factible para estos tipos de aplicaciones.

## 1.5.2 Herramienta de modelado

### Visual Paradigm 8.0 para el Lenguaje de Modelado Unificado

Es una herramienta CASE destinada a apoyar el ciclo de vida de un software, principalmente mediante el diseño de diagramas. Permite la generación automática de reportes, reconocimiento de artefactos de ingeniería a partir de reconocimiento de textos formales o informales; implementa una actualización automática del modelo de diseño y código además, permite optimizar la descripción textual de elementos de código a partir de la descripción visual. Es capaz de importar y exportar elementos de otras herramientas CASE como Rational Rose y presenta una interfaz de uso intuitiva y con excelentes facilidades a la hora de modelar los diagramas. (10)

## 1.5.3 Metodología de desarrollo

Por las características del proyecto y la posibilidad de contar con un cliente fuertemente involucrado con el proyecto, así como la posibilidad de trabajo con ambientes visuales y reusabilidad (SDK de Microsoft SAPI 5 para la síntesis de voz), además de la visible modularidad existente, se decidió adoptar la metodología RAD para la investigación.

### **RAD se aconseja utilizar cuando:** (11)

- La aplicación funcionará de manera independiente.
- Se pueden usar mayormente bibliotecas existentes.
- Desempeño y confiabilidad no crítica.
- Distribución limitada, interna o vertical.
- Alcance del proyecto limitado.
- El sistema puede dividirse en muchos módulos independientes.
- El producto está dirigido a un mercado altamente especializado.
- El proyecto cuenta con fuertes limitantes de tiempos parciales (*timeboxes*).
- La tecnología requerida tiene más de un año en el mercado.

### **No es aconsejable utilizarlo cuando** (11)

# Capítulo 1. Fundamentación teórica

- Existen pocos componentes reutilizables.
- El desarrollo no puede aprovechar herramientas de alto nivel.
- RAD se convierta en QADAD (Rápido y Sucio Desarrollo Ágil, por sus siglas en inglés).
- Métodos RAD para desarrollar sistemas operativos (confiabilidad demasiado alta) o juegos (desempeño demasiado alto).
- Existen riesgos técnicos de tecnología de punta.
- El producto pone en riesgo la misión o la vida.
- El producto no puede ser modularizado.

## **Ventajas de RAD (11)**

- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Visibilidad temprana.
- Mayor flexibilidad e involucramiento de los usuarios.
- Menor codificación manual.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.
- Interfaz gráfica estándar.

## **Desventajas de RAD (11)**

- Costo de herramientas integradas y equipo necesario.
- Progreso más difícil de medir.
- Menor precisión científica.
- Funciones reducidas.

## **Descripción de la metodología RAD**

# Capítulo 1. Fundamentación teórica

Rapid Application Development (RAD), Desarrollo Rápido de Aplicaciones, es un integrado conjunto de técnicas, directrices y herramientas que facilita la solución de los requisitos del software de un cliente en un corto periodo de tiempo. Este predefinido espacio de tiempo es llamado “Caja de Tiempo”. El producto de software no sale al final del ciclo de desarrollo, mejor aún, evoluciona durante el proceso de desarrollo RAD, basándose en una constante retroalimentación de parte del cliente. (12)

Cada incremento o segmento de la aplicación toma normalmente de tres a nueve meses. El proceso RAD utiliza un equipo pequeño con altas habilidades individuales (incluyendo a los clientes) para trabajar juntos usando herramientas que aceleren las pruebas, los prototipos y la construcción de artefactos del producto de software. (13)

Generalmente los equipos están compuestos por alrededor de seis personas, incluyendo desarrolladores y usuarios de tiempo completo del sistema, así como aquellas personas involucradas con los requisitos. Los desarrolladores han de ejercer los roles de: analista, diseñador y programador.

Esta metodología propone las siguientes fases (14):

**Inicialización:** Esta fase define el alcance general del proyecto, la estructuración de los temas de trabajo y selecciona las partes interesadas pertinentes para iniciar el proyecto. Se centra en formalizar los objetivos, el alcance y validar los requisitos del proyecto. Esta fase representa aproximadamente el 6 % de los proyectos subvencionados.

La fase de inicialización informa a los involucrados de las limitaciones de un proyecto RAD. Después de una breve inmersión en el área funcional, el propietario, los coordinadores y los especialistas, analizan y presentan las limitaciones del método y en un plan de acción, incluyen un plan de comunicación. Durante esta fase se identifican los involucrados, se buscan soluciones, se evalúan los medios necesarios y se llega a un acuerdo entre el cliente y el especialista para el lanzamiento del proyecto.

**Enmarcado:** En esta fase se realiza la especificación y recogida de requisitos, donde los usuarios expresan sus necesidades durante las entrevistas de grupo. Tiene el objetivo de formalizar los requisitos, establecer el plan de enmarcación del enmarcado de funciones, así como la validación de los objetivos y las limitaciones por el cliente y sus expertos. Esta fase representa alrededor del 9 % del proyecto.

**Diseño:** Tiene como objetivo definir la solución y objetivo en términos de modelos, para generar un primer prototipo. Los usuarios también están involucrados en este paso y participan en la refinación y validación de modelos de organización. También validan el prototipo de primer nivel con la ergonomía y la cinemática generales de la aplicación. Se espera entre 4 y 8 días de sesiones por comisión. Esta fase representa alrededor del 23 % del proyecto.

# Capítulo 1. Fundamentación teórica

**Construcción:** Su objetivo es desarrollar la aplicación a partir de las funcionalidades entregadas. Durante esta fase, el equipo RAD debe construir la aplicación por módulos. El usuario-cliente sigue participando activamente en la definición detallada y validación permanente de prototipos. Se necesitan varias sesiones iterativas. Esta fase representa aproximadamente el 50 % del proyecto.

**Finalización:** Sistemas parciales se han obtenido en la fase anterior, es en esta fase donde se formaliza una ejecución global del sistema final, se produce la documentación del usuario y se transfiere el conocimiento a los entrenadores para realizar la capacitación. Esta fase representa alrededor del 12 % del proyecto.

La siguiente figura muestra las fases por las que transita un proyecto utilizando la metodología de desarrollo ágil RAD:

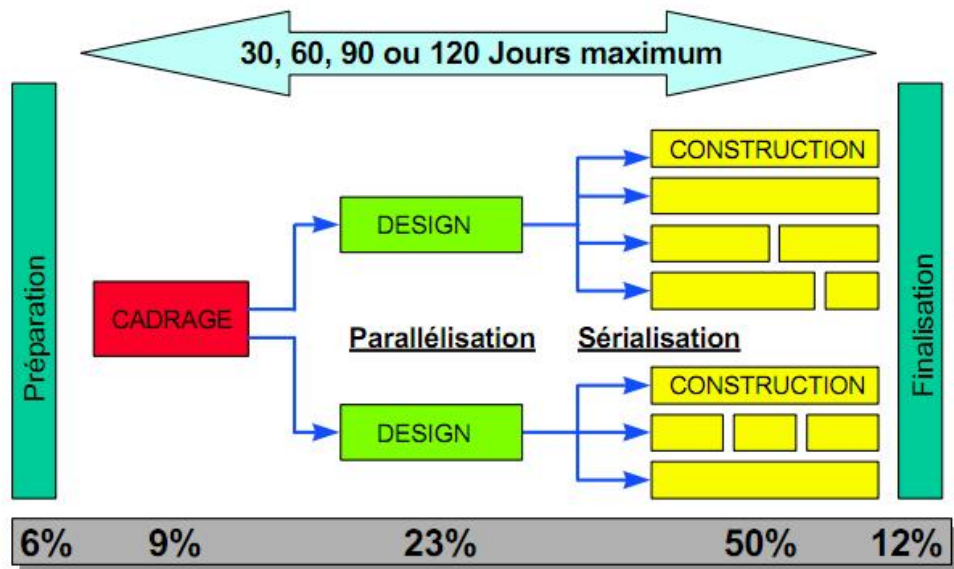


Figura 2 Fases del modelo de Desarrollo Rápido de Aplicaciones.

## 1.5.4 Tecnología de síntesis de voz

La síntesis de voz o *Text To Speech* (TTS), es el proceso mediante el cual se busca que una máquina genere voz, o simule el habla humana, a partir de un texto, mediante el uso de procesos matemáticos y el estudio de la fonética y la teoría lingüística. (15)

Una de las aplicaciones más importantes está relacionada con aspectos de accesibilidad, en los que se busca que sea útil a personas con discapacidades lectoras sobre sistemas informáticos.

La calidad de una voz sintética vendrá dada por:

# Capítulo 1. Fundamentación teórica

- Su inteligibilidad: ¿con qué facilidad/dificultad es entendida?
- Su naturalidad: ¿en qué medida se asemeja a la voz *real* de un humano?

## **Tipos de Síntesis de voz:**

Existen diferentes tecnologías de síntesis de voz. (15)

### **Sintetizador Articulatorio:**

Se basa en hacer una analogía entre los parámetros de los órganos articulatorios del hombre, con parámetros circuitales, que le permitan a la máquina emularlos. La calidad que ofrecen es muy alta, sin embargo es un sistema muy complicado en términos de controlar y configurar los parámetros del sistema. Los principales problemas que presenta este tipo de síntesis, es que hace falta mucha información sobre los movimientos de los articulados, y se requieren de una gran carga computacional.

### **Sintetizadores por formantes:**

Busca imitar el tracto vocal humano mediante una serie de filtros, que se activan por fuentes que simulan las cuerdas vocales. Estos se basan en modelos matemáticos que controlan la frecuencia fundamental o la amplitud de la onda, además de otros factores como el tiempo de apertura del sonido o lo abrupto de la onda, entre otros, que pueden afectar la forma en que sale la voz.

### **Sintetizadores por concatenación de forma de onda:**

Tienen una base de datos con unidades o fonemas pregrabados, de modo que después del correcto análisis, éste concatena las unidades adecuadas para generar nuevas frases. Su grado de complejidad es alto y se limita a las características de un solo hablante (aquel que realiza las grabaciones), pero mejora la calidad y se tienen muy buenos resultados, siendo el método más usado en la actualidad para diversas aplicaciones.

### **Síntesis por concatenación:**

Como ya se mencionó anteriormente, este método está basado en bloques de voz pregrabada de forma tal, que dependiendo del sistema, se unan adecuadamente para poder llegar a formar las frases o palabras que se ingresaron por texto.

## **Bloques**

Los bloques son la base de este sistema de síntesis de voz, pues a partir de los cuales, se buscará que la computadora construya las frases a generar. Existen diferentes maneras de generarlos a partir de

# Capítulo 1. Fundamentación teórica

grabaciones de una persona que mantenga una forma constante al hablar. Estas grabaciones para obtener la información, pueden realizarse de 4 formas diferentes:

- Frases naturales: Se graban diferentes frases que contengan las unidades que se quieren.
- Frases portadoras: Se graba una estructura fija, donde se cambian algunas palabras para obtener los bloques que se quieren.
- Palabras aisladas: se graba cada palabra, y se procede a fragmentarla si es necesario.
- Logatomos: palabras que no tengan sentido semántico, pero que contengan el bloque deseado. Es generalmente usado cuando el bloque consiste en **difonemas** o **trifonemas**, y tiene una estructura dada.

En cualquier tipo de grabación se debe tener claro cuál será el punto de corte.

En cada sistema de síntesis de voz se debe tener claro cuál va a ser la unidad básica de concatenación, si serán **sílabas**, **palabras**, **frases**, etc. Esto se hace basado en la calidad que se busca al generar la voz y en la cantidad de memoria con la que se cuenta, pues a tramos de voz más grandes mejor calidad, pero también se necesita mayor uso de memoria. No se recomienda usar palabras, pues la entonación de las palabras es totalmente diferente al usarlas en frases que al decir las de forma independiente, se pierden características tales como, el ritmo o la entonación, además, se necesitaría una gran base de datos para agrupar todas las palabras existentes. Por otro lado existe la síntesis por sílabas, lo cual también necesitaría un gran número de las mismas.

La unidad más aceptada hasta el momento es el difonema, que sería la unión de fonemas. El lenguaje y su fono táctica, tienen una gran influencia en la definición de los **difonemas**, por lo que el número puede variar dependiendo del idioma que se maneje en el sintetizador. Para el español existe un aproximado de 900 **difonemas**. Estos consisten en crear los bloques desde la mitad de un fonema a la mitad del siguiente, sin embargo al tomar los fonemas como el bloque base, se presentan problemas articulatorios en las palabras.

Una ventaja de los difonemas, es que no se usan todos, por lo que se pueden eliminar los formados por dos consonantes que no se den en el lenguaje. En resumen, se puede obtener una base de datos con 550 **difonemas**, incluyendo los necesarios para representar las diferentes acentuaciones, y algunos casos con trifonemas (unión de tres fonemas) que se digan a gran velocidad. Estos se graban, se seleccionan puntos de corte y se deben etiquetar.

# Capítulo 1. Fundamentación teórica

Para resumir, al realizar la elección de bloques del sistema se debe tener en cuenta 4 criterios importantes:

- Usar el menor número de concatenación posible
- Que las uniones se realicen en zonas que sean acústicamente estables
- Que limitaciones de memoria se tiene.
- Tener en cuenta el control de la coarticulación.

La síntesis denominada por concatenación de difonemas, es la que se emplea en esta investigación. Se proponen las voces de Jorge (España), Esperanza (México) y Diego (Argentina). Desdichadamente la variante cubana no se encuentra editada hasta el momento de esta investigación.

## Conclusiones parciales.

Después de realizar un análisis de los fundamentos teóricos asociados con el problema a resolver, se pueden obtener las siguientes conclusiones:

- La sistematización de los elementos teóricos abordados, tanto desde el punto de vista informático, como lingüístico y pedagógico, permitieron encontrar las vías y herramientas factibles para concebir la solución del problema de esta investigación.
- Que existe una tipología de software educativo que es la denominada “Entrenador” (“**Skill and practice**”) a la cual se subordina el software objeto de estudio en esta tesis y que en ocasiones no está contemplada dentro de las clasificaciones más difundidas.
- En nuestro país existen algunos entrenadores ortográficos con un alcance muy limitado en cuanto a exposición de contenidos y empleo de recursos que sean consecuentes con el método viso-audio-gnósico-motor, sustento automatizable de la metodología de la enseñanza de la ortografía.
- El logro de un entrenador ortográfico con “carácter abierto” en el que se pueda emplear cualquier palabra de la lengua, siendo el lenguaje natural un sistema débilmente estructurado y altamente versátil con una disminución del costo intelectual de implementación, plantea como reto la codificación de algoritmos que permitan inferir aspectos de carácter lingüísticos no triviales, como:
  - La división de palabras en sílabas.

# Capítulo 1. Fundamentación teórica

- La determinación de fenómenos lingüísticos versátiles como la existencia de diptongos, hiatos, acentos diacríticos, y enclíticos, así como la clasificación de las palabras acorde con su acentuación gráfica, teniendo en cuenta las excepciones de la regla.



# Capítulo 2. Presentación de la propuesta de solución

## Capítulo 2. Presentación de la propuesta de solución

### Introducción

Acorde con las características del problema a resolver, el tiempo de desarrollo y los recursos humanos, resulta necesario el empleo de una metodología ágil. La metodología RAD integra el desarrollo iterativo y la construcción de prototipos vinculados con productos con interfaz gráfica de usuario y entornos de desarrollo integrados completos, como es el caso de *LiveCode*, herramienta definida para realizar la aplicación. Tradicionalmente RAD, tiende a englobar la usabilidad, reusabilidad, utilidad y la rapidez de ejecución, utilizando un enfoque de construcción basado en componentes. Genera artefactos como casos de uso, modelos conceptuales de comunicación y de datos, además, requiere clientes en perenne comunicación con el proceso de desarrollo de software.

### 2.1 Inicialización

Se realizó el primer encuentro formal con el cliente donde éste expone sus necesidades, objetivos que persigue y las principales funciones y características que desea cumpla el futuro sistema informático. Argumentando además, el impacto que puede tener el desarrollo de este tipo de aplicación.

La información recopilada sirvió para realizar una propuesta del futuro sistema. En el siguiente esquema (Figura 3), se refleja la situación planteada por el cliente y su posible solución.

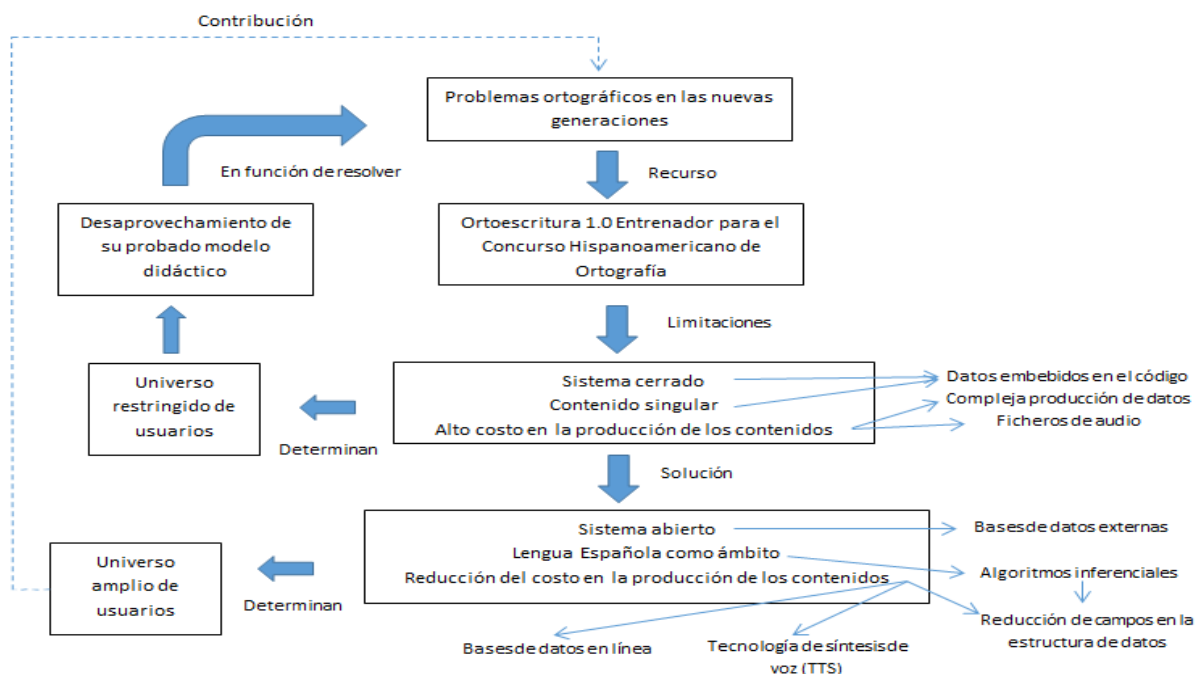


Figura 3 Diagrama del problema del cliente.

# Capítulo 2. Presentación de la propuesta de solución

## 2.1.1 Propuesta de sistema

Se propone la creación de una aplicación de escritorio con las siguientes características:

- Los procedimientos metodológicos utilizados en el entrenador preexistente se preservan.
- Habilitar la lectura de datos a partir de bases de datos externas con exclusivamente dos campos palabra/significado y oración-frase/autor. El resto de los campos eliminados serán inferidos mediante la implementación de los algoritmos inferenciales de orden lingüísticos.
- Reemplazo de los ficheros de audio por mecanismos de síntesis de voz
- Garantía de la integridad pedagógica mediante la implementación del acceso a una base de datos en línea.

## 2.2 Enmarcado funcional

En el enmarcado funcional se recogen y formalizan los requisitos tanto funcionales como no funcionales de la aplicación, estableciéndose una jerarquía de funcionalidades para el establecimiento de prioridades a la hora de implementar. El cliente solicitó quedar al tanto del avance del desarrollo de los requisitos; para esto se utilizó el documento de seguimiento de requisitos de la aplicación (Ver Tabla 4); a su vez se planteó emplear los casos de uso como el artefacto para detallar las funcionalidades.

### 2.2.1 Descripción de las funcionalidades

A continuación se describen los requisitos funcionales y no funcionales obtenidos para el futuro sistema.

#### Requisitos funcionales (RF)

**RF1:** Registrar usuario en el sistema: El usuario, para poder comenzar a utilizar la aplicación, ha de introducir su nombre, apellido y seleccionar el tipo de voz (voz sintética con la que el sistema interactuará con el usuario en lo adelante), como parte de su registro en el sistema.

**RF1.1:** Crear traza: Se crea la traza del usuario utilizando el nombre, apellido, fecha y hora actual, como parte del identificador de una traza que recogerá las estadísticas de todos los ejercicios que realice.

**RF2:** Configurar modalidades de entrenamientos: Luego del registro del usuario en el sistema, se listan las bases de datos disponibles para realizar los entrenamientos, agrupándolos en dos modalidades:

- Significado – Palabra: contienen diversas palabras (agrupadas según el nivel de dificultad o temática) con su respectivo significado.
- Oración – Frase: con disímiles frases u oraciones agrupadas por materias.

## Capítulo 2. Presentación de la propuesta de solución

Además, muestra la forma en que pueden presentarse los ejercicios de entrenamiento: secuencial o al azar; así como el tipo de entrenamiento que se puede realizar de acuerdo a la modalidad que se seleccione.

RF2.1: Listar bases de datos locales: Se listan los datos (temáticas disponibles) de las bases de datos locales para realizar los entrenamientos.

RF2.2: Listar bases de datos en línea: Se listan los datos (temáticas disponibles) de las bases de datos que se encuentren en el servidor para realizar los entrenamientos.

RF3: Realizar entrenamiento de tipo “Familiarización y análisis”: Entrenamiento para la modalidad significado-palabra, donde se presentan listados de palabras para que el usuario interactúe con ellas, escuche su pronunciación, lea su significado y de desearlo, analice la palabra.

RF4: Realizar entrenamiento de tipo “Dictado e inferencia”: Entrenamiento para la modalidad significado-palabra donde se presentan palabras que deberán ser bien escritas. La palabra será dictada y se mostrará el significado de esta como auxilio para la solución del ejercicio. Se muestran datos y estadísticas relacionadas con el ejercicio y las opciones específicas del tipo de ejercicio: pista, palabra, escuchar, reglas, ayuda.

RF5: Realizar entrenamiento de tipo “Escritura con apoyo”: Entrenamiento para la modalidad oración-frase donde se presentan oraciones o frases que deberán ser completadas correctamente por el usuario. El completamiento del ejercicio se puede realizar mediante edición libre, escribiendo grafemas y colocando tildes, mayúsculas y signos de puntuación. De ser necesario se puede restablecer la oración o frase en cuestión. Se muestran datos y estadísticas relacionadas con el ejercicio y las opciones específicas del tipo de ejercicio: ver frase, escuchar, reglas, ayuda.

RF6: Realizar entrenamiento de tipo “Escritura libre”: Entrenamiento para la modalidad significado-palabra donde se dictan oraciones o frases que deberán ser escritas correctamente por el usuario. Se muestran datos y estadísticas relacionadas con el ejercicio y las opciones específicas del tipo de ejercicio: ver frase, escuchar, reglas, ayuda.

RF7: Mostrar opciones del ejercicio: Diversas opciones que se presentan en los ejercicios de entrenamiento para apoyar y ayudar al usuario. Las opciones se muestran de acuerdo al tipo de ejercicio, estas son:

- Pista: Muestra una versión procesada de la palabra donde se ocultan los accidentes ortográficos.
- Palabra: Muestra la palabra de forma parpadeante durante un segundo.

## Capítulo 2. Presentación de la propuesta de solución

- Escuchar: Reproduce la palabra, frase u oración correspondiente.
- Ver frase: Muestra la frase en un cuadro de diálogo.
- Reglas: Muestra las reglas ortográficas.
- Ayuda: Muestra los elementos necesarios para el entendimiento y desarrollo del ejercicio de entrenamiento.

RF8: Realizar análisis de palabra: Funcionalidad encargada de tomar una palabra y analizarla ortográficamente en materia de acentuación gráfica.

RF8.1: Dividir palabra en sílabas: Divide una palabra en sílabas.

RF8.2: Detectar sílaba tónica: Localiza la fuerza de pronunciación de una palabra.

RF8.3: Clasificar tipo de palabra: Clasifica una palabra en aguda, llana, esdrújula o sobreesdrújula.

RF8.4: Determinar regularidad lingüística: Determina si una palabra tiene hiato, diptongo y sus tipos (fuerte con débil y débil con débil), acento diacrítico o enclítico y su terminación.

RF9: Consultar trazas: Muestra todas las trazas existentes permitiendo seleccionar la deseada y comprobar los resultados y datos de los ejercicios que el usuario realizó como parte del entrenamiento.

RF9.1: Listar trazas: Lista las trazas disponibles.

RF9.2: Mostrar traza: Muestra los datos de la traza seleccionada.

### **Requisitos no funcionales (RNF)**

RNF1: Usabilidad: La aplicación puede ser utilizada por cualquier persona con conocimientos básicos de informática.

RNF2: Restricciones de diseño:

- Sistema operativo: Windows.
- Tipo de aplicación: Escritorio.
- Herramienta de desarrollo: *LiveCode* Community 6.1.
- Lenguaje de programación: *Transcript*.
- Herramienta de modelado UML: Visual Paradigm v8.0

RNF3: Interfaz: Su interfaz ha de ser ergonómicamente parecida a la aplicación predecesora.

# Capítulo 2. Presentación de la propuesta de solución

## RNF4: Requisitos de hardware.

- Requisitos mínimos:
  - Periféricos: Teclado, ratón.
  - Video: 64MB.
  - Velocidad: Procesador 512MHz.
  - Memoria RAM: 512MB
  - Espacio de almacenamiento: 8MB
- Requisitos recomendados:
  - Periféricos: Teclado, ratón.
  - Video: 128MB.
  - Velocidad: Procesador Intel Pentium 4 a 1GHz o similares.
  - Memoria RAM: 1GB
  - Espacio de almacenamiento: 8MB
  - Conexión por módem o banda ancha para los datos en línea.

## **2.2.2 Jerarquía de requisitos del futuro sistema**

### **1. Indispensable**

- RF1: Registrar usuario en el sistema.
- RF2: Configurar modalidades de entrenamientos.
- RF2.1: Listar bases de datos locales.
- RF3: Realizar entrenamiento de tipo “Familiarización y análisis”.
- RF4: Realizar entrenamiento de tipo “Dictado e inferencia”.
- RF5: Realizar entrenamiento de tipo “Escritura con apoyo”.
- RF6: Realizar entrenamiento de tipo “Escritura libre”.
- RF8: Realizar análisis de palabra.
- RF8.1: Dividir palabra en sílabas.

# Capítulo 2. Presentación de la propuesta de solución

- RF8.2 Detectar sílaba tónica.
- RF8.3 Clasificar tipo de palabra.
- RF8.4 Determinar regularidad lingüística.

## 2. Futura Obligación

- RF1.1 Crear traza.
- RF7: Mostrar opciones del ejercicio.
- RF9: Consultar traza.
- RF9.1 Listar trazas.
- RF9.2 Mostrar traza.

## 3. Mejoras Significativas

- RF2.2 Listar bases de datos en línea.

## 4. Mejoras útiles

- No se detectaron funcionalidades para este nivel jerárquico.

## 5. Confort

- No se detectaron funcionalidades para este nivel jerárquico.

## 2.3 Casos de uso

Los casos de uso son los encargados de modelar el sistema desde el punto de vista del usuario. Según Pressman, el objetivo de estos es: “definir los requisitos funcionales y operativos del sistema (producto), diseñando un escenario de uso acordado por el usuario final, y el equipo de desarrollo; proporcionar una descripción clara y sin ambigüedades de como el usuario final interactúa con el sistema y viceversa, y proporcionar una base para la validación de las pruebas.” (16)

### 2.3.1 Patrones de casos de uso

Para el modelado de casos de uso, se utilizan plantillas para describir la estructura y organización que debieran presentar los casos de uso, éstos son los llamados patrones de casos de uso y son los encargados de describir la interacción entre el usuario y el sistema con un poco más de detalle.

## Capítulo 2. Presentación de la propuesta de solución

Un patrón de casos de uso utilizado fue el patrón “Extend” que consta de dos casos de uso con una relación de extensión entre ellos, donde el caso de uso extensión, puede ser aplicado en el caso de uso extendido bajo ciertas condiciones. Se aplica cuando un flujo puede extender el flujo de otro caso de uso, así como ser realizado en sí mismo. (17) La siguiente figura representa el diagrama de casos de uso del sistema.

### 2.3.2. Diagrama de casos de uso

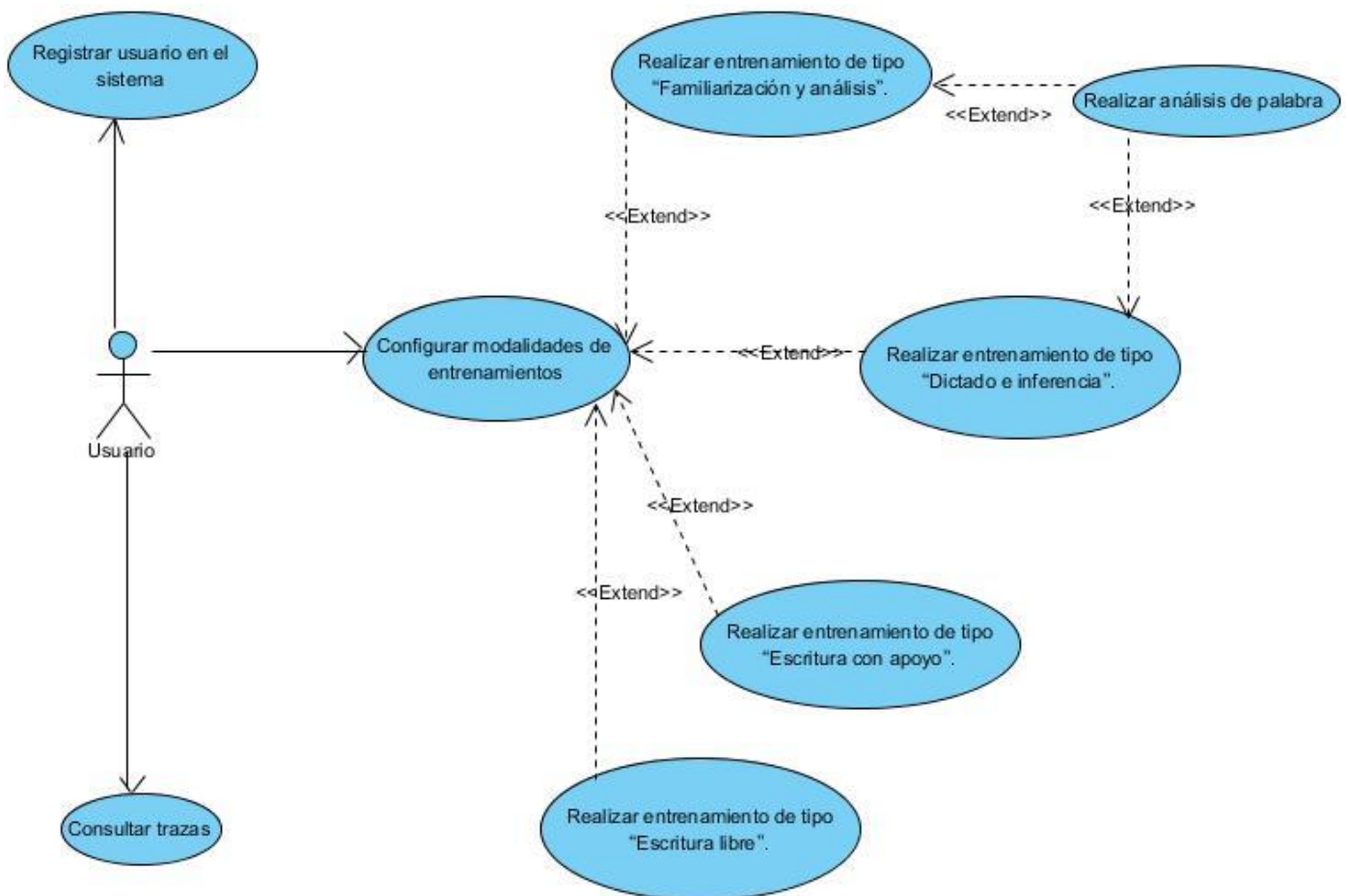


Figura 4 Diagrama de casos de uso de la aplicación.

### 2.3.3 Descripción de casos de uso

Tabla 3 Descripción del CU Configurar modalidades de entrenamientos.

<b>Objetivo</b>	Configurar la forma en que desea realizar determinado ejercicio
<b>Actores</b>	Usuario

## Capítulo 2. Presentación de la propuesta de solución

<b>Resumen</b>	Se listan las bases de datos disponibles para realizar los entrenamientos agrupándolos en dos modalidades: Significado-Palabra, contienen diversas palabras (agrupadas según el nivel de dificultad o temática) con su respectivo significado; Oración-Frase, con disimiles frases u oraciones agrupadas por materias. Muestra la forma en que desea se presenten los ejercicios de entrenamiento: Secuencial o al azar; además del tipo de entrenamiento que se quiere realizar de acuerdo a la modalidad que se seleccione.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Debe haberse seleccionado el tipo de entrenamiento "Escritura libre" de la modalidad "oración-frase" con una forma de presentación.	
<b>Postcondiciones</b>	Se realizan o no, el o los ejercicios. Se guardan los datos en la traza del usuario	
<b>Flujo de eventos</b>		
<b>Flujo básico: Configurar modalidades de entrenamientos</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El caso de uso se inicia cuando el actor se registra.	
2.		<p>Muestra las temáticas que se encuentran en la base de datos local y permite:</p> <ul style="list-style-type: none"> <li>• Seleccionar la opción de cargar las temáticas de una base de datos en línea.</li> </ul> <p>Además brinda la posibilidad de:</p> <ul style="list-style-type: none"> <li>• Seleccionar ejercicios de la variante Oración-Frase.</li> <li>• Sección 1: Seleccionar ejercicios de la variante Significado-Palabra. Ver sección.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Regresar</li> <li>• Ver traza</li> </ul>
3.	Selecciona ejercicios de la modalidad Oración-Frase.	
4.		Brinda la posibilidad de seleccionar los niveles:



## Capítulo 2. Presentación de la propuesta de solución

		<ul style="list-style-type: none"> <li>• Familiarización y análisis</li> <li>• Dictado e inferencia</li> </ul> <p>Y la forma de presentación:</p> <ul style="list-style-type: none"> <li>• Secuencial</li> <li>• Aleatoria</li> </ul> <p>Además permite:</p> <ul style="list-style-type: none"> <li>• Comenzar</li> </ul>
5.	Selecciona Familiarización y análisis.	
6.	Selecciona la forma de presentación.	
7.	Introduce los datos.	
8.	Selecciona Comenzar.	
9.		Valida los datos.
10.		Muestra la pantalla para realizar el ejercicio. <i>Ver CU Realizar ejercicios del tipo Familiarización y análisis.</i>
11.		El caso de uso termina.
<b>Flujos alternos 5.1 El actor selecciona Dictado e inferencia</b>		
	<b>Actor</b>	<b>Sistema</b>
5.1.1	Selecciona dictado e inferencia.	
5.1.2	Introduce los datos.	
5.1.3	Selecciona comenzar.	
5.1.4		Valida los datos.
5.1.5		Muestra la pantalla para realizar el ejercicio. <i>Ver CU Realizar ejercicios del tipo Dictado e inferencia.</i>
		El caso de uso termina.
<b>Flujos alternos 3.1 El actor selecciona la opción “Ver trazas”.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1.1		Muestra la pantalla con las trazas del usuario. <i>Ver CU Consultar trazas.</i>
3.1.2		Regresa al paso 2 del flujo básico.
<b>Flujos alternos 3.2 El actor selecciona la opción “Regresar”.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.2.1		Va al <i>CU Registrar usuario en el sistema</i>
3.2.2		El caso de uso termina

## Capítulo 2. Presentación de la propuesta de solución

Flujos alternos 9.1 Datos incorrectos		
	Actor	Sistema
9.1.1		Muestra un mensaje.
9.1.2		Regresa al paso 2 del flujo básico.
Relaciones	CU Incluidos	
	CU Extendidos	

Tabla 4 Sección Seleccionar ejercicios de la modalidad Significado-Palabra

Flujo básico: Seleccionar ejercicios de la modalidad Significado-Palabra		
	Actor	Sistema
1	Selecciona ejercicios de la modalidad Significado-Palabra.	
2		Brinda la posibilidad de seleccionar los niveles <ul style="list-style-type: none"> <li>• Escritura con apoyo</li> <li>• Escritura libre</li> </ul> Y la forma de presentación con campos para introducir números: <ul style="list-style-type: none"> <li>• Secuencial</li> <li>• Aleatoria</li> </ul>
3	Selecciona Escritura con apoyo.	
4	Selecciona la forma de presentación.	
5	Introduce los datos.	
6	Selecciona Comenzar.	
7		Valida los datos.
8		Va al <u>CU Realizar ejercicios del tipo Escritura con apoyo</u>
9		El caso de uso termina
Flujos alternos 3.1 El actor selecciona Escritura libre		
	Actor	Sistema
3.1.1	Selecciona la forma de presentación.	
5.1.2	Introduce los datos.	
5.1.3	Selecciona comenzar.	
5.1.4		Valida los datos.

## Capítulo 2. Presentación de la propuesta de solución

5.1.5	Muestra la pantalla para realizar el ejercicio. <i>Ver CU Realizar ejercicios del Escritura libre.</i>
	El caso de uso termina.



Figura 5 Interfaz del CU Configurar modalidades de entrenamiento.

### 2.4 Diseño de la aplicación

Al terminar la captura de requisitos se realizó una reunión donde participó el cliente y aprobó formalmente los requisitos de la aplicación, acordando entonces los entregables que se iban a realizar en la fase de diseño: Modelo Conceptual de Comunicación, Modelo Conceptual de Datos y el prototipo inicial.

#### 2.4.1 Modelo Conceptual de Comunicación

El Modelo Conceptual de Comunicación ofrece una visión conjunta de las relaciones entre los objetos. Con este modelo se pretende representar los objetos implicados en la implementación de una función de la aplicación, simbolizando solo los objetos que participan en esta comunicación. (18) En el siguiente diagrama (Figura 6) se muestran los objetos relacionados en el *CU Realizar análisis de palabra*.

## Capítulo 2. Presentación de la propuesta de solución

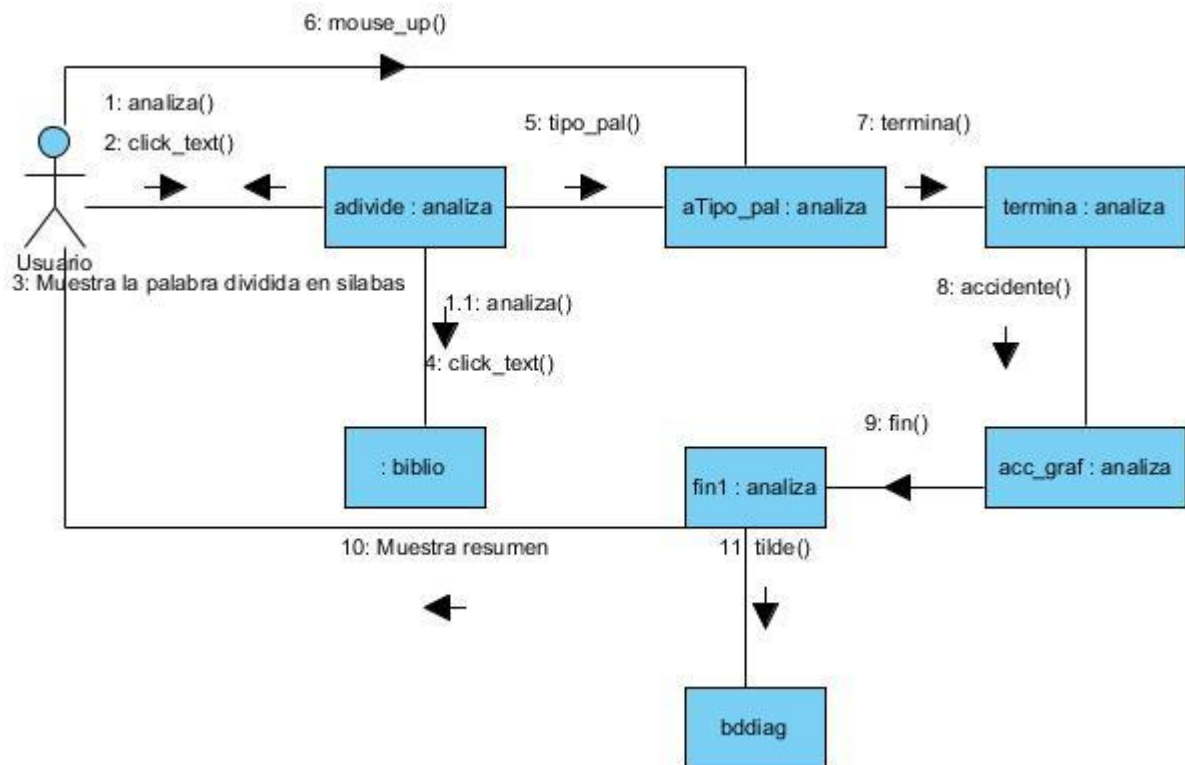


Figura 6 Modelo conceptual de comunicación del CU Realizar análisis de palabra.

### 2.4.2 Modelo Conceptual de Datos

Es una representación estructural de datos fácilmente comprensible para describir el sistema de información a través de entidades. (19) De forma general estos datos sirven para describir la estructura de la base de datos. En esta aplicación se tienen dos bases de datos correspondientes a los contenidos: una que se encuentra en un servidor central que contiene la tabla "*tematica*" y la otra que se encuentra de manera local en forma de ficheros, representados por las tablas "*frases*" y "*pals*". En la Figura 7 se representan estas bases de datos.

# Capítulo 2. Presentación de la propuesta de solución

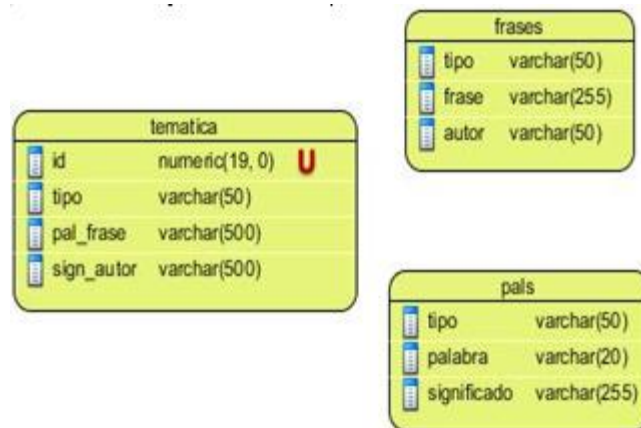


Figura 7 Estructura de las bases de datos referente a los contenidos

## 2.5 Prototipo inicial

Para este punto del ciclo de vida del software, el usuario ha de recibir un prototipo inicial de la aplicación. Este prototipo inicial, presenta funcionalidades limitadas pero se hace énfasis en la interfaz de usuario para brindar una vista previa al cliente del funcionamiento y flujo de la aplicación. El flujo o representación de este nivel básico del sistema se representa en la Figura 8 mostrada a continuación:

# Capítulo 2. Presentación de la propuesta de solución

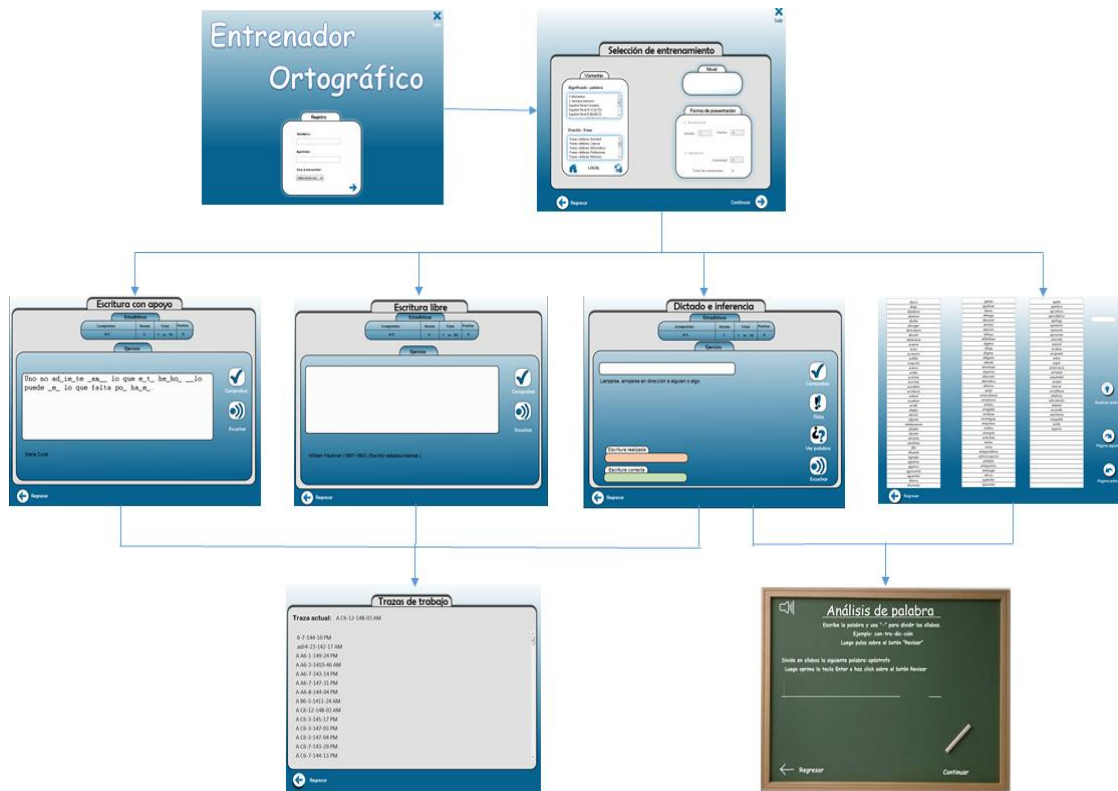


Figura 8 Representación de las interfaces del prototipo inicial

## Conclusiones parciales

Con la culminación de este capítulo se ha presentado la propuesta de solución a la problemática planteada arribándose a las siguientes conclusiones:

- Acorde con la metodología utilizada se desarrolló un prototipo inicial de la aplicación con las funcionalidades indispensables y todas las interfaces, el cual fue aprobado por el cliente.
- El empleo de la metodología RAD, resultó viable para estructurar, planificar y controlar el proceso de desarrollo de la nueva versión del entrenador, quedando como valor añadido de la investigación, un conjunto de artefactos que podrán ser utilizados en sus futuras mejoras.
- La base de datos resultante para este producto es extremadamente sencilla en comparación con su predecesor en virtud de los datos que ahora son inferidos, y las operaciones que se realizan sobre estos datos se limitan exclusivamente a la carga de la información (no hay búsqueda, ordenamiento), por lo que se decidió mantener ficheros en formato ASCII encriptados para garantizar su seguridad.

# Capítulo 3. Implementación y pruebas

## Capítulo 3 Implementación y pruebas

### Introducción

Los artefactos obtenidos durante la enmarcación y diseño son la base para la construcción de la aplicación. Durante el presente capítulo, se abordan aspectos del proceso de implementación de los requisitos funcionales identificados, además del proceso de validación del software implementado mediante diferentes tipos de pruebas.

### 3.1 Construcción

El prototipo inicial obtenido en la fase de diseño es la base de la construcción de la aplicación. En esta fase se implementan todas las funcionalidades sobre este primer prototipo generado, en lo que se denomina “refinación del prototipo”. El producto en evolución transita por un proceso de “liberación permanente”, donde se entrega continuamente una versión actualizada del prototipo que ha de aprobar el cliente. Los momentos para realizar estas entregas fueron acordados, definiéndose 4 etapas con sus condiciones de liberación:

- Etapa 1: Concluir los casos de uso: Registrar usuario en el sistema, Configurar modalidades de entrenamiento y Consultar traza.
- Etapa 2: Concluir los casos de uso referentes a los 4 tipos de entrenamientos.
- Etapa 3: Concluir el caso de uso Realizar análisis de palabra.
- Etapa 4: Funcionamiento total de la aplicación.

El seguimiento de esta implementación se registró en el documento “Seguimiento de requisitos de la aplicación” solicitado por el cliente, el cual se muestra en la siguiente imagen:

## Capítulo 3. Implementación y pruebas

Ortoescritura 2.0						
Seguimiento de requisitos de la aplicación						
Nº	Fecha de la solicitud	Fecha de inicio	Fecha de integración.	Fecha de entrega	Identificación del requisito	Carga
1	29/1/2014	14/3/2014	17/3/2014	23/3/2014	RF1: Registrar usuario en el sistema	Indispensable
2	29/1/2014	16/3/2014	17/3/2014	23/3/2014	RF1.1: Crear traza	Futura obligación
3	29/1/2014	18/3/2014	22/3/2014	23/3/2014	RF2: Configurar modalidades de entrenamientos	Indispensable
4	29/1/2014	20/3/2014	22/3/2014	23/3/2014	RF2.1: Listar bases de datos locales	Indispensable
5	29/1/2014	20/3/2013	22/3/2014	23/3/2014	RF2.2: Listar bases de datos en línea	Mejora significativa
6	29/1/2014	24/3/2014	3/4/2014	4/4/2014	RF3: Realizar entrenamiento de tipo "Familiarización y análisis"	Indispensable
7	29/1/2014	25/3/2014	3/4/2014	4/4/2014	RF4: Realizar entrenamiento de tipo "Dictado e inferencia"	Indispensable
8	29/1/2014	30/3/2014	3/4/2014	4/4/2014	RF5: Realizar entrenamiento de tipo "Escritura con apoyo":	Indispensable
9	29/1/2014	2/4/2014	3/4/2014	4/4/2014	RF6: Realizar entrenamiento de tipo "Escritura libre":	Indispensable
10	29/1/2014	3/4/2014	3/4/2014	4/4/2014	RF7: Mostrar opciones del ejercicio	Futura obligación
11	29/1/2014	7/4/2014	24/4/2014	25/4/2014	RF8: Realizar análisis de palabra	Indispensable
12	29/1/2014	10/4/2014	24/4/2014	25/4/2014	RF8.1: Dividir palabra en sílabas	Indispensable
13	29/1/2014	16/4/2014	24/4/2014	25/4/2014	RF8.2: Detectar sílaba tónica	Indispensable
14	29/1/2014	17/4/2014	24/4/2014	25/4/2014	RF8.3: Clasificar tipo de palabra	Indispensable
15	29/1/2014	21/4/2014	24/4/2014	25/4/2014	RF8.4: Determinar regularidad lingüística	Indispensable
16	29/1/2014	16/3/2014	17/3/2014	23/3/2014	RF9: Consultar trazas	Futura obligación
17	29/1/2014	16/3/2014	17/3/2014	23/3/2014	RF9.1: Listar trazas	Futura obligación
18	29/1/2014	16/3/2014	17/3/2014	23/3/2014	RF9.2: Mostrar traza	Futura obligación

Figura 9 Seguimiento de requisitos de la aplicación.

Durante la refinación del prototipo y hasta la entrega final de éste, se efectúan determinadas pruebas para comprobar la calidad y su buen funcionamiento.

### 3.2 Pruebas de software

Las pruebas de software es una de las etapas fundamentales para asegurar la calidad de la aplicación donde se realiza una revisión final de las especificaciones, diseño e implementación. En este proceso se ejecutan pruebas dirigidas a los componentes del software y/o a todo el sistema de software mediante el uso de técnicas de prueba.



## Capítulo 3. Implementación y pruebas

Para el desarrollo de las pruebas se utilizó una técnica de planificación propuesta por la metodología RAD denominada “**Hitos cero defectos**”; que permite controlar el avance de la aplicación en materia de visibilidad y calidad. Se basa en el planteamiento de puntos de control en la aplicación, asegurando que, hasta el momento en que se define el punto, no exista ningún error en ella. Esta técnica se ejecuta durante el proceso de refinamiento del prototipo, efectuándose las siguientes pruebas:

**Pruebas de unidad:** Encargadas de probar el correcto funcionamiento de un módulo. Durante esta prueba se han de comprobar las interfaces del módulo, estructura de los datos, las condiciones límite, así como los caminos de ejecución. En este sistema, estas pruebas se efectúan para comprobar el módulo que se está desarrollando en cada etapa de la “liberación permanente”. De esta manera, se establece por cada iteración, un hito de cero defectos.

**Pruebas de integración:** Encargadas de construir la estructura del programa. Tienen el objetivo de asegurar que los módulos comprobados, mediante las pruebas de unidad, funcionen correctamente en conjunto. Establece el hito de cero defectos al realizar la integración de los módulos del prototipo en cada iteración.

**Pruebas de validación:** Se enfoca en detectar errores en los requisitos, sobre todo en elementos que son necesarios para el usuario final. La validación se consigue cuando el software funciona de acuerdo con las expectativas razonables del cliente. En este punto, el prototipo se ha terminado completamente y se realizan pruebas por parte del desarrollador y del cliente, buscando la aceptación del producto.

### 3.2.1 Métodos de prueba

**Pruebas de caja blanca:** Son las encargadas de verificar el código interno del sistema Su uso posibilita la obtención de casos de prueba que garantizan, al menos una vez, que todos los caminos independientes de cada módulo sean ejecutados. Además, facilita ejercitar todas las decisiones lógicas y permite la ejecución de cada ciclo.

**Pruebas de caja negra:** Las pruebas de caja negra se centran en comprobar todos los requisitos funcionales del sistema. En otras palabras, permite al ingeniero de software realizar un conjunto de operaciones y condiciones que comprueben todos los requisitos funcionales del sistema para lograr la operatividad de las funciones, entradas y salidas correctas e integridad de la información externa. Este tipo de prueba se enfoca en encontrar errores en las siguientes categorías (16):

- Funciones incorrectas o ausentes.
- Errores de interfaz.

## Capítulo 3. Implementación y pruebas

- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

En la validación de la solución propuesta en la presente investigación, se hizo uso de este tipo de prueba utilizando el método de partición equivalente, donde se dividen los campos de entrada del programa en clases de datos, de los que se pueden derivar casos de prueba.

Cabe acotar que el cliente, para el desarrollo de las pruebas, propició un juego de palabras que representaban todos los posibles casos de acentuaciones y asegurarían el correcto funcionamiento del análisis de la palabra (CU Realizar análisis de palabra) en la aplicación.

Tabla 5 Lista de palabras tipificadas.

ratón	país	Salud
jugó	Raúl	Queso
parís	destruía	sucesivamente
árbol	rápidamente	Fui
esdrújula	más	La
María	aéreo	comióselo
	Camagüey	

### 3.2.2 Descripción de los Casos de Prueba

**Caso de prueba:** Registrar usuario en el sistema

**Descripción general:** El caso de uso se inicia, cuando el usuario inicia la aplicación. Para poder comenzar a utilizar la aplicación, se ha de introducir su nombre y apellido como parte de su registro en el sistema. Estos datos son guardados junto a la fecha y hora actual como parte del identificador de una traza que recogerá las estadísticas de todos los ejercicios que realice. Se selecciona el tipo de voz que en lo adelante interactuará con el usuario

**Condiciones de ejecución:** Debe haberse iniciado la aplicación

Tabla 6 Registrar usuario en el sistema.

Escenario	Descripción	Nombre	Apellidos	Voz (Seleccione voz)	Respuesta del sistema	Flujo central
EC 1.1						Iniciar aplicación

## Capítulo 3. Implementación y pruebas

	El caso de uso se inicia cuando el usuario inicia la aplicación				Se muestran en la pantalla los campos nombre, apellidos y la opción seleccionar voz. Y permite: <ul style="list-style-type: none"> <li>• Registrarse</li> <li>• Salir</li> </ul>	
EC 1.2	Introduce los datos nombre y apellidos	V	V	N/A		Iniciar aplicación
EC 1.3	Selecciona la voz "Jorge"	V	N/A	V	Recibe, hablándole al usuario con la voz de Jorge.	Iniciar aplicación
EC 1.4	Selecciona la opción "Registrarse"	V	V	V	Verifica los datos. Termina el caso de prueba	Iniciar aplicación/Registrar
EC 1.5	Los datos están incorrectos	I I I V V V	I I V V I I V	I V I V I	Muestra un mensaje informando que los datos están incorrectos	Iniciar aplicación/Registrar

## Capítulo 3. Implementación y pruebas

		V	V	V		
EC 1.6	El usuario selecciona la opción salir	N/A	N/A	N/A		

Tabla 7 Variables del CU Registrar usuario en el sistema

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre	Cadena	NO	Cadena de caracteres donde la primera letra ha de ser mayúscula. No se permiten números
2	Apellido	Cadena	NO	Cadena de caracteres donde la primera letra ha de ser mayúscula. No se permiten números
3	Tipo de voz	Cadena	NO	Jorge, Diego, Esperanza

### Pruebas de aceptación

Una vez que el desarrollador realizó todas las pruebas, se realizan las de aceptación. Estas son aquellas que realiza el cliente directamente sobre las funcionalidades del software, con el fin de determinar su nivel de satisfacción respecto a la implementación de los requisitos funcionales y no funcionales definidos. Los aspectos principales en los que se basaron estas pruebas fueron:

- La entrada y salida de datos de las funcionalidades, centrándose en el área del análisis de las palabras.
- El correcto flujo de interfaces.
- El uso de diferentes bases de datos.

### 3.2.3 Resultados obtenidos

Cada etapa de liberación trajo consigo las no conformidades o anomalías de diferente índole. La siguiente imagen muestra el resultado de las pruebas en la etapa 3.

## Capítulo 3. Implementación y pruebas

Orotescritura 2.0						
Seguimiento de tratamiento de anomalías						
N° Anomalia	Fecha descubierto	Descripción de la disfuncionalidad	Índice de gravedad	Tiempo transcurrido	Fecha de solución	Estado de la anomalía
1	22/04/2014	No se dirige a las tarjetas referentes al hiato y diptongo	Bajo	1h	22/04/2014	Resuelta
2	22/04/2014	No abre el stack "análisis"	Bajo	1h	22/04/2014	Resuelta
3	22/04/2014	No utiliza todas las palabras seleccionadas por el usuario	Alto	48h	24/04/2014	Resuelta
4	22/04/2014	Realiza de forma incorrecta la adición y substracción de puntos	Bajo	1h	22/04/2014	Resuelta
5	22/04/2014	No reconoce la sílaba tónica de la palabra	Alto	24h	23/04/2014	Resuelta
6	22/04/2014	No divide correctamente las palabras con "x" intervocálica	Alto	24h	23/04/2014	Resuelta
7	23/04/2014	No reconoce en todos los casos el acento diacrítico	Alto	2h	23/04/2014	Resuelta
8	23/04/2014	No divide correctamente las palabras con "h" intermedia	Alto	3h	23/04/2014	Resuelta

Figura 10 Seguimiento de tratamiento de anomalías de la etapa 3.

### 3.3 Finalización

Una vez terminada la aplicación, se le hace la entrega formal del sistema validado al cliente. El sistema final está constituido por 5 pilas o “**stacks**” conformadas por una o varias tarjetas o “**cards**” que contienen el código y/o las interfaces que componen el producto desarrollado:

- **Stack “biblio”**: Es referenciada como la biblioteca de código de la aplicación, es decir, almacena la mayoría del código utilizado. Los métodos y funciones desarrollados en esta pila pueden ser utilizados en cualquier momento por cualquier otra tarjeta o pila.
- **Stack “entrenador”**: Es la pila principal de la aplicación. Contiene la mayoría de las interfaces y funcionalidades desarrolladas.
- **Stack “familiar”**: Debido a la especificidad de uno de los requisitos funcionales Realizar entrenamiento de tipo “Familiarización y análisis”, se utilizó esta pila para ese propósito.
- **Stack “análisis”**: Se encarga de una sola funcionalidad: “Análisis de palabra” y sigue diferentes recorridos de tarjetas de acuerdo a los datos recibidos.
- **Stack “bddiag”**: Contiene una serie de tarjetas con contenido referente a las reglas ortográficas.

Para una mejor comprensión de la estructura de las pilas del producto final, se muestra el siguiente diagrama:

## Capítulo 3. Implementación y pruebas

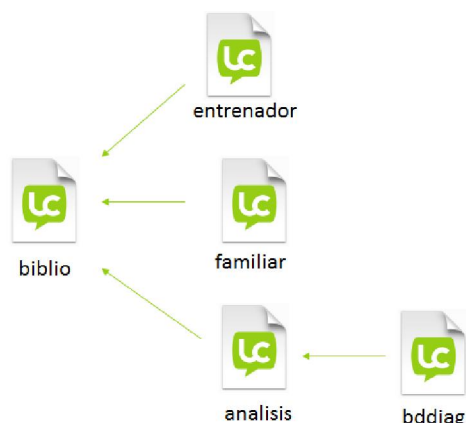


Figura 11 Estructura de los Stacks de la aplicación final

Para una mejor comprensión del funcionamiento del sistema final, se muestra la siguiente figura:



Figura 12 Arquitectura del funcionamiento del sistema.

### Conclusiones parciales

- El desarrollo de la aplicación evidenció la dificultad de los algoritmos empleados, sobre todo en la elaboración de funciones y métodos para el análisis de las palabras.
- La herramienta de desarrollo **LiveCode**, resultó óptima para el desarrollo del sistema, propiciando funciones nativas de alta utilidad en el trabajo con textos, direcciones y audio.
- Las pruebas efectuadas demostraron que la conversión de Ortoescritura en un sistema abierto, permitió aprovechar su modelo didáctico, garantizando que éste trabaje con cualquier palabra de la Lengua Española y con ello facilitar, que cualquier experto o docente, se pueda convertir en autor de contenidos del entrenador, para los escenarios lingüístico más diversos.

## Conclusiones generales

- La externalización de los datos, la reducción de su estructura y la introducción de la tecnología de síntesis de voz, permitió reducir al mínimo, el costo intelectual e informático de la producción de contenidos de entrenamiento.
- La implementación de un acceso en línea para el entrenamiento, garantiza un escenario pedagógicamente íntegro, que compensa el riesgo que se asume con el carácter abierto que posee la nueva versión del entrenador.
- Aunque no constituyó un objetivo explícito de esta investigación, las mejoras introducidas a Ortoescritura, dada la ampliación del universo de usuarios potenciales, permitirá diversificar los escenarios de su uso y con ello contribuir al mejoramiento de la ortografía en múltiples sectores de la población.

## Recomendaciones

La investigación realizada y en particular la aplicación de una metodología de desarrollo, permitió vislumbrar aristas del entrenador, susceptibles para ser mejoradas:

- Crear un módulo que registre la traza de varios competidores conectados a la base de datos en línea, para implementar una variante de concursos a distancia.
- Introducir un SGBD como SQLite para mejorar las prestaciones de la recogida de los resultados del proceso de entrenamiento, con la intención de introducir métodos estadísticos en los análisis de las trazas.
- Introducir un diccionario de la RAE (Real Academia Española), articulado al proceso de entrenamiento, para con ello, insertar un componente de alto valor agregado al entrenador.



## Referencias bibliográficas

1. Sánchez, Jaime. *Construyendo y aprendiendo con el computador*. Santiago de Chile : s.n., 1999.
2. Marquéz, Pere. *El software educativo*. 1996.
3. Rizzo, César Labañino. *Multimedia en la Educación*. La Habana : Pueblo y Educación, 2003.
4. Portal Educativo Cubaeduca. [En línea] [Citado el: 14 de 3 de 2014.] <http://www.cubaeduca.cu>.
5. Entrenador de Ortografía I . [En línea] [Citado el: 13 de 2 de 2014.] [http://educarteoax.com/pedagogizando/index.php?option=com\\_content&view=article&id=180](http://educarteoax.com/pedagogizando/index.php?option=com_content&view=article&id=180).
6. Entrenador de Ortografía II. [En línea] [Citado el: 13 de 2 de 2014.] [http://educarteoax.com/pedagogizando/index.php?option=com\\_content&view=article&id=193](http://educarteoax.com/pedagogizando/index.php?option=com_content&view=article&id=193).
7. Tordera, Juan Carlos. *El abecé de la lingüística computacional*. s.l. : Arco-Libros, 2011.
8. Rodríguez, Liván. *Revolución Multimedia*. s.l. : Centro de Estudios Visofted, 2005.
9. *LiveCode. Discover compile-free coding, the fasterway to develop*.
10. Visual-paradigm. [En línea] [Citado el: 2014 de 2 de 7.] <http://www.visual-paradigm.com/support/documents/vpuserguide.jsp>.
11. *Rapid Application Development*. Maner, Walter. 1997.
12. *RAD realities: beyond the hype to how rad really works*. Gottesdiener, Ellen. 1995.
13. RAD. [En línea] [Citado el: 16 de 4 de 2014.] <http://www.rad.fr>.
14. *Méthode RAD. Éléments fondamentaux*. Vickoff, Jean-Pierre.
15. Loeber, María Angélica. *Síntesis de la voz y del habla*. Buenos Aires : s.n., 2011.
16. Pressman, Richard. *Ingeniería del Software. Un enfoque práctico*. 5ta.
17. Övergaard, Gunnar y Palmkvist, Karin . *Use Cases Patterns and Blueprints*. 2004.
18. Pamela, Contreras. *Diagrama de comunicación UML 2.0*. Tecnicatura Universitaria en Informática.
19. MERISE. [En línea] [Citado el: 5 de 6 de 2014.] <http://www.commentcamarche.net/contents/659-merise-modele-conceptuel-des-donnees>.

## **Bibliografía consultada**

BALLESTA, Javier. Educar para los medios en una sociedad multicultural. 2009.

AREA, Manuel. Introducción a la Tecnología Educativa. 2009.

CABERO, Julio. Tecnología educativa. 2008

MARTIN, James. *Design of man-computer dialogues*. 1973.

VICKOFF, Jean-Pierre. RAD et nouvelle économie: un même combat. 2001

VICKOFF, Jean-Pierre. RADicalisation strategique des projects. 2001.

VICKOFF, Jean-Pierre. Le RAD: une nouvelle gestion de projet. 1996.

VICKOFF, Jean-Pierre. Communication: Merise, le RAD et l'Objet. 2000.

VICKOFF, Jean-Pierre. Les approches modernes d'évaluation. 2005.

LARMAN, Craig. UML y patrones. Introducción al análisis y diseño orientado a objetos. 1999.

BOYLE, T. Design for Multimedia Learning 1997.

VIRVOU, Maria, et. al. Combining Software Games with Education: Evaluation of its Educational Effectiveness. 2005.

## Glosario de términos

**Modelo didáctico de un software educativo:** Se entiende por modelo didáctico de un software educativo al conjunto de atributos que se definen para lograr los procesos de enseñanza o aprendizaje y lo caracterizan como medio didáctico.

**Usuarios potenciales de un software educativo:** En el diseño pedagógico de un software educativo o guion del software coincide con el concepto de Público al que va dirigido el software. Este público se condiciona por componentes del proceso de enseñanza-aprendizaje entre los que figuran los objetivos, los contenidos y los métodos.

**Costo intelectual de un software educativo:** En el proceso de autoría de un software educativo se considera costo intelectual del software al nivel de esfuerzo intelectual que requiere la producción de sus contenidos. Resulta obvio que se trata de una medida subjetiva, dependiente de las competencias del contenidista, sin embargo es constatable que en contextos específicos existen tareas más complejas que otras.

**Viso-audio-gnósico-motor:** Se trata de un método de la didáctica de la Ortografía que se sustenta en el empleo de recursos visuales, auditivos, conceptuales y motrices para enseñar y aprender ortografía. Viso-leer, Audio-escuchar, gnósico-comprender significado, motor-escribir.

**SDK:** kit de desarrollo de software o SDK (siglas en inglés de Software Development Kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etc.

**Widget:** En el contexto de la programación de aplicaciones visuales, los widgets tienen un significado más amplio como componente o control visual que el programador reutiliza, suelen reunirse varios widgets en juegos de herramientas de widgets. Los programadores los usan para construir interfaces gráficas de usuario (GUI).

**Curva de aprendizaje:** Una curva de aprendizaje describe el grado de éxito obtenido durante el aprendizaje en el transcurso del tiempo. Es un diagrama en que el eje horizontal representa el tiempo transcurrido y el eje vertical el número de éxitos alcanzados en ese tiempo.

## Definiciones alternativas

Curva de aprendizaje en el uso coloquial a veces usado: una curva empinada señala: "el tema es difícil de aprender".

Junto a la definición académica existe en el lenguaje uso coloquial un uso diametralmente diferente del término. Especialmente en el mercadeo de software y en la rama de herramientas se dice que una curva de aprendizaje es empinada cuando el aprendizaje del uso de una nueva herramienta o programa es difícil y largo. Visto así, una curva de aprendizaje plana significa un aprendizaje fácil y eficiente. En tal diagrama estarían representados en el eje horizontal la acumulación de lo aprendido y en el eje vertical la acumulación del tiempo gastado. La pendiente de la curva es en ese caso la razón de tiempo a avance (tiempo/avance) y es como lo define Wright. La diferencia entre ambas consiste en que la definición académica representa al aprendizaje como el éxito obtenido y la definición coloquial lo representa por el esfuerzo invertido.

**Hiperentorno de Aprendizaje:** Se define como una modalidad informática que se sustenta en la tecnología hipermedia y en el que están presentes un conjunto de elementos representativos de diversas tipologías de software educativo.

**Anexos**

**Anexo 1 Encuesta**

1- Encuesta aplicada a los estudiantes de preuniversitario del curso 2012-2013 en el marco del concurso nacional de ortografía que se llevó a cabo en los días 11-15 del mes de noviembre del 2013 en la escuela Vladimir I. Lenin de la capital.

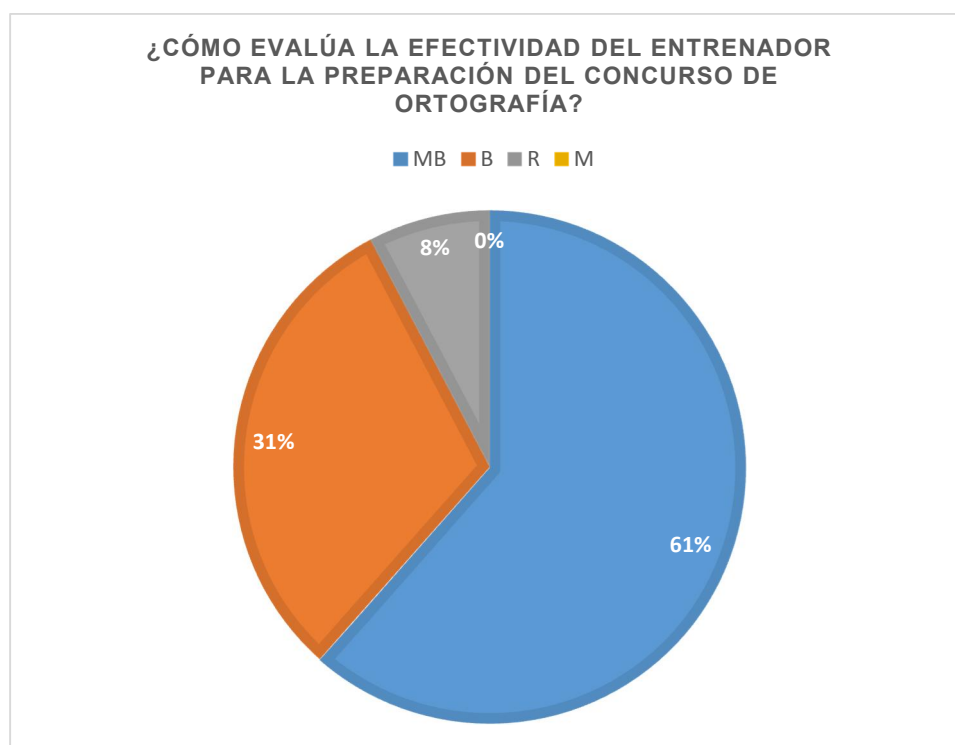
a) ¿Cómo evalúa la efectividad del entrenador para la preparación del concurso de Ortografía?

Muy bueno: 32 estudiantes

Bueno: 16 estudiantes

Regular: 4 estudiantes

Malo: 0 estudiantes

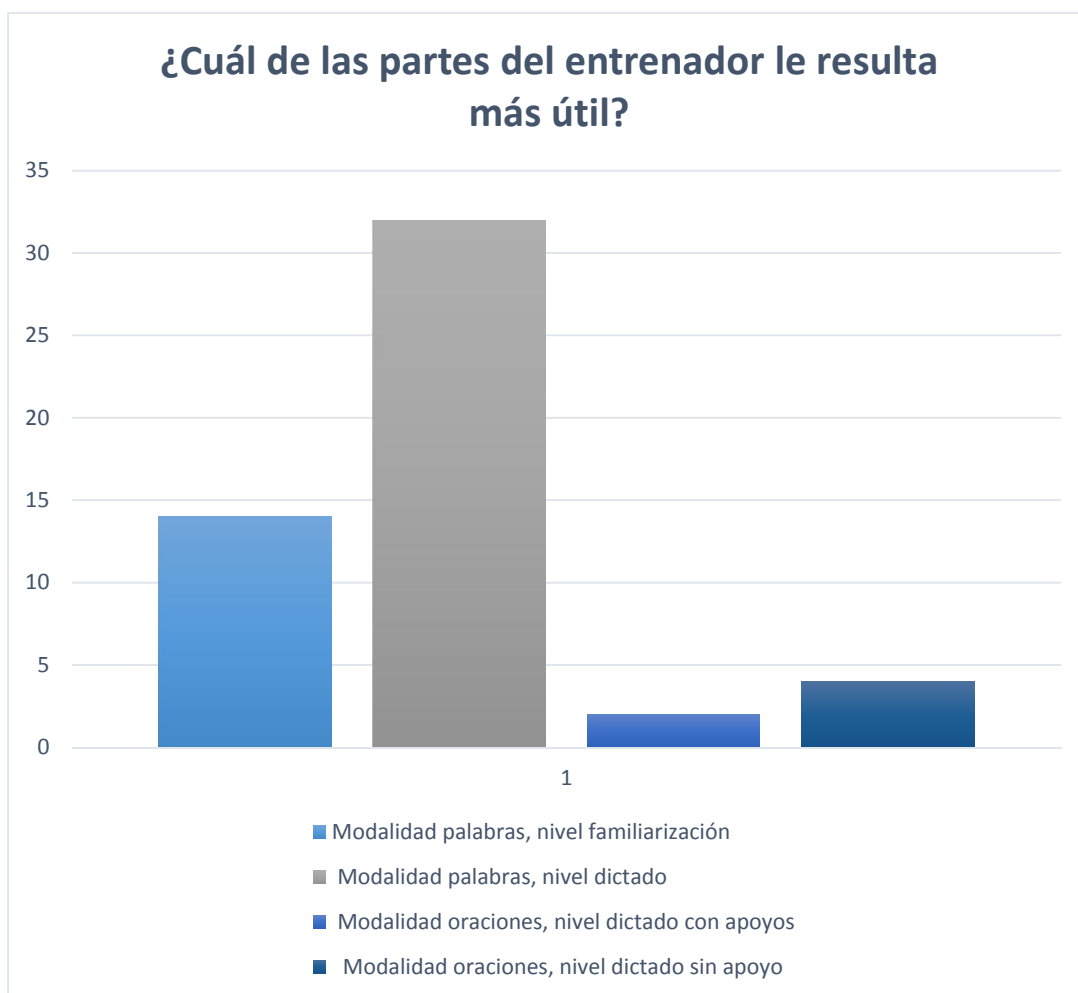


**Figura 13 Efectividad del entrenador**

b) ¿Cuál de las partes del entrenador le resulta más útil?

a) Modalidad palabras, nivel familiarización: 14 estudiantes.

- b) Modalidad palabras, nivel dictado: 32 estudiantes.
- c) Modalidad oraciones, nivel dictado con apoyo: 2 estudiantes.
- d) Modalidad oraciones, nivel dictado sin apoyo: 4 estudiantes.



**Figura 14 Parte más útil del entrenador**

c) ¿Qué le gustaría que tuviera un próximo entrenador?

Que pueda trabajar con los conceptos y terminología de las asignaturas: 16 estudiantes.

Que se le puedan incluir nuevas palabras u oraciones: 28 estudiantes.

Que se trabaje por parejas: 2 estudiantes.

Que existan premios si se logran altas puntuaciones: 6 estudiantes.

Población:

52 estudiantes (16 competidores y 36 estudiantes de la Lenin).

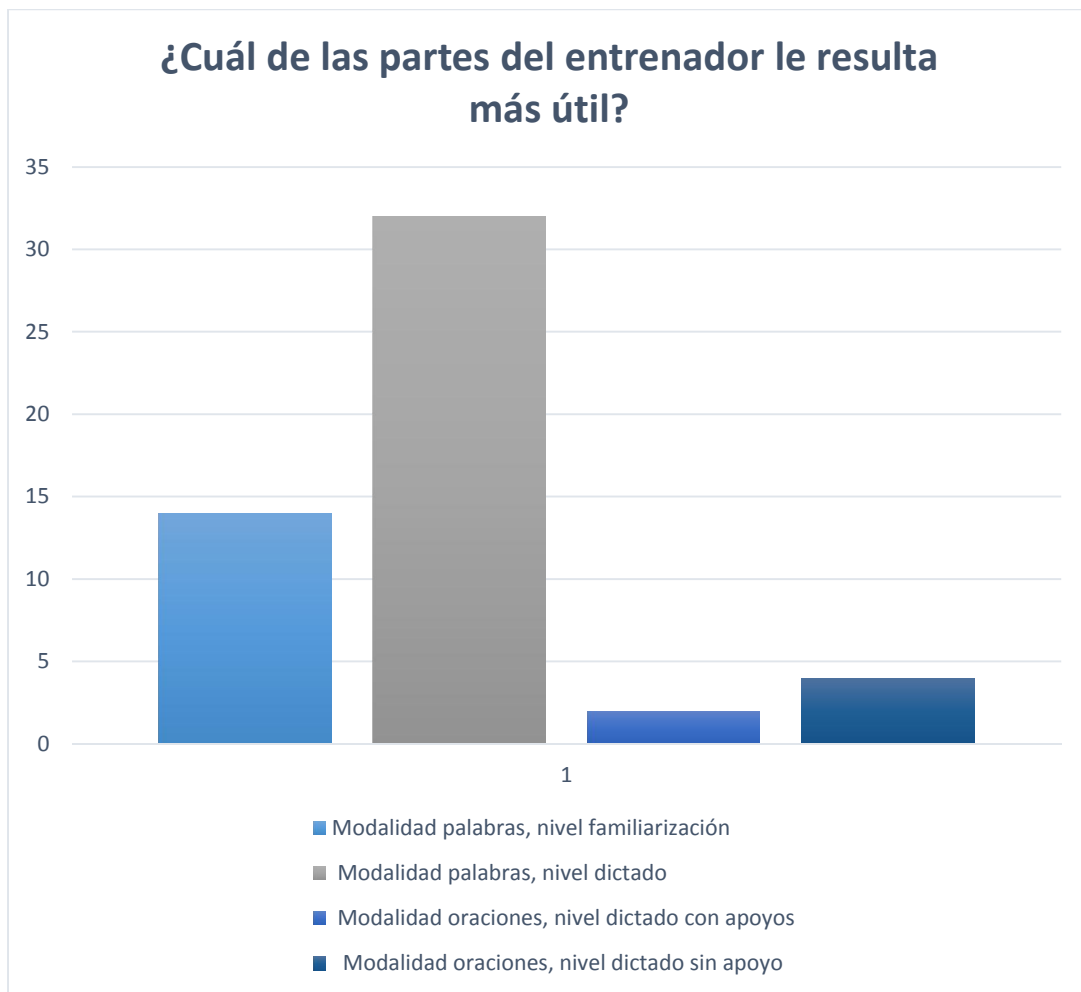


Figura 15 Parte más útil del entrenador

## Anexo 2 Descripciones de Casos de Uso

### CU Realizar análisis de palabra

Tabla 8 Descripción del CU Realizar análisis de palabra

<b>Objetivo</b>	Analizar las palabras que el usuario desee
<b>Actores</b>	Usuario
<b>Resumen</b>	Funcionalidad encargada de tomar una palabra y analizarla ortográficamente en materia de acentuación gráfica.

<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Debe haberse seleccionado “Analizar palabra”	
<b>Postcondiciones</b>	Resumen general de la palabra	
<b>Flujo de eventos</b>		
<b>Flujo básico: Analizar palabra</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El caso de uso se inicia cuando el actor selecciona la opción Analizar palabra	
2.		Brinda la posibilidad de dividir la palabra en sílabas y permite <ul style="list-style-type: none"> <li>• Comprobar</li> <li>• Regresar</li> <li>• Ayuda</li> </ul>
3.	Divide la palabra en sílabas	
4.		Inserta un “-” en cada espacio entre palabras
5.	Selecciona la opción “Comprobar”	
6.		Valida los datos. Son correctos
7.		.Muestra y reproduce (voz sintética) un mensaje.
8.		Brinda la posibilidad de seleccionar la sílaba tónica
9.	Selecciona la sílaba tónica	
10.		Valida la respuesta. Es correcta



11.		Muestra y reproduce (voz sintética) un mensaje.
12.		Permite <ul style="list-style-type: none"> <li>• Continuar</li> </ul>
13.	Selecciona "Continuar"	
14.		Va a la sección " <i>Tipo de palabra</i> " de este caso de uso
15.		El caso de uso termina
<b>Flujos alternos 10.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
10.1.1		Muestra y reproduce (voz sintética) un mensaje.
10.1.2		Va al paso 9 del flujo básico
<b>Flujos alternos 8.1 Valida los datos. Son incorrectos</b>		
4.5.1		Muestra y reproduce (voz sintética) un mensaje.
4.5.2		Va al paso 3 del flujo básico
<b>Flujos alternos 4.7 El actor selecciona la opción "Ayuda".</b>		
4.7.1		Ver <u>CU Mostrar opciones del ejercicio. Sección Ayuda</u>
4.7.2		El caso de uso termina
<b>Flujos alternos 4.8 El actor selecciona la opción "Terminar".</b>		
4.8.1		Muestra el mensaje ¿Desea abandonar?" y permite: <ul style="list-style-type: none"> <li>• Si</li> </ul>

		• No
4.8.2	Selecciona "Si"	
4.8.3		<p>Si viene del <u>CU Familiarización y análisis</u></p> <p>Va al <u>CU Familiarización y análisis</u></p> <p>Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u></p> <p>Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u></p>
4.8.4		El caso de uso termina
<b>Flujos alternos 4.8.2 El actor selecciona la opción "No".</b>		
4.8.2.1		Vuelve al paso 2 del flujo básico.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Mostrar opciones de ejercicios.

## Sección "Tipo de palabra"

Tabla 9 Sección "Tipo de palabra"

Flujo básico: Tipo de palabra		
	Actor	Sistema
1		<p>Brinda la posibilidad de seleccionar el tipo de palabra:</p> <ul style="list-style-type: none"> <li>• Aguda</li> </ul>

		<ul style="list-style-type: none"> <li>• Llana</li> <li>• Esdrújula</li> <li>• Sobresdrújula</li> </ul> <p>Y permite</p> <ul style="list-style-type: none"> <li>• Terminar</li> </ul>
2	Selecciona "Aguda"	
3		Valida la respuesta. Es correcta
4		Muestra y reproduce (voz sintética) un mensaje.
5		Permite <ul style="list-style-type: none"> <li>• Continuar</li> </ul>
6	Selecciona "Continuar"	
7		<p>Si la palabra tiene hiato va a la <u>sección "Hiato"</u> de este caso de uso</p> <p>Si tiene diptongo va a la <u>sección "Diptongo"</u> de este caso de uso</p> <p>Si no va a la <u>sección "terminación"</u> de este caso de uso</p>
8		El caso de uso termina
<b>Flujos alternos 3.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico

Flujos alternos 2.1 Selecciona “Llana”		
2.		Valida la respuesta. Es correcta
4		Muestra y reproduce (voz sintética) un mensaje.
5		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>
6	Selecciona “Continuar”	
7		Si la palabra tiene hiato va a la <u>sección “Hiato”</u> de este caso de uso  Si tiene diptongo va a la <u>sección “Diptongo”</u> de este caso de uso  Si no va a la <u>sección “terminación”</u> de este caso de uso
8		El caso de uso termina
Flujos alternos 2.2 Selecciona “Esdrújula”		
2.2. 1		Valida la respuesta. Es correcta
2.2. 2		Muestra y reproduce (voz sintética) un mensaje.
2.2. 3		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>
2.2. 4	Selecciona “Continuar”	

<p>2.2. 5</p>		<p>Si la palabra tiene hiato va a la <u>sección “Hiato”</u> de este caso de uso</p> <p>Si tiene diptongo va a la <u>sección “Diptongo”</u> de este caso de uso</p> <p>Si no va a la <u>sección “terminación”</u> de este caso de uso</p>
<p>2.2. 7</p>		<p>El caso de uso termina</p>
<p><b>Flujos alternos 2.3 Selecciona “Sobresdrújula”</b></p>		
<p>2.3. 1</p>		<p>Valida la respuesta. Es correcta</p>
<p>2.3. 3</p>		<p>Muestra y reproduce (voz sintética) un mensaje.</p>
<p>2.3. 4</p>		<p>Permite</p> <ul style="list-style-type: none"> <li>• Continuar</li> </ul>
<p>2.3. 5</p>	<p>Selecciona “Continuar”</p>	
<p>2.3. 6</p>		<p>Si la palabra tiene hiato va a la <u>sección “Hiato”</u> de este caso de uso</p> <p>Si tiene diptongo va a la <u>sección “Diptongo”</u> de este caso de uso</p>

		Si no va a la <u>sección "Terminación"</u> de este caso de uso
<b>2.3.</b> <b>7</b>		El caso de uso termina
<b>Flujos alternos 2.1 El actor selecciona la opción "Terminar".</b>		
<b>2.1.1</b>		Muestra el mensaje ¿Desea abandona?" y permite: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
<b>2.1.2</b>	Selecciona "Si"	
<b>2.1.3</b>		Si viene del <u>CU Familiarización y análisis</u> Va al <u>CU Familiarización y análisis</u>  Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u> Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u>
<b>4.8.4</b>		El caso de uso termina

**Tabla 10 Sección Hiato**

<b>Flujo básico: Diptongo</b>		
	<b>Actor</b>	<b>Sistema</b>
<b>1</b>		Muestra la pregunta ¿Esta palabra tiene diptongo?

		<p>Y Brinda la posibilidad de seleccionar de responder:</p> <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul> <p>Y permite</p> <ul style="list-style-type: none"> <li>• Terminar</li> </ul>
2	Selecciona "Si"	
3		Valida la respuesta. Es correcta
4		Muestra y reproduce (voz sintética) un mensaje.
5		Permite
		<ul style="list-style-type: none"> <li>• Continuar</li> </ul>
6	Selecciona "Continuar"	
7		Va a la <u>sección "terminación"</u> de este caso de uso
8		El caso de uso termina
<b>Flujos alternos 3.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.1 Selecciona "No"</b>		
2.		Valida la respuesta. Es correcta
4		Muestra y reproduce (voz sintética) un mensaje.

5		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>
6	Selecciona "Continuar"	
7		Va al paso 7 del flujo básico
<b>Flujos alternos 3.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.1 El actor selecciona la opción "Terminar".</b>		
2.1.1		Muestra el mensaje "¿Desea abandonar?" y permite: <ul style="list-style-type: none"> <li>Si</li> <li>No</li> </ul>
2.1.2	Selecciona "Si"	
2.1.3		Si viene del <u>CU Familiarización y análisis</u> Va al <u>CU Familiarización y análisis</u>  Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u> Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u>



Tabla 11 Sección diptongo

Flujo básico: Hiato		
	Actor	Sistema
1		Muestra la pregunta ¿Esta palabra tiene hiato? Y Brinda la posibilidad de seleccionar de responder: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul> Y permite <ul style="list-style-type: none"> <li>• Terminar</li> </ul>
2	Selecciona "Si"	
3		Valida la respuesta. Es correcta
4		Muestra y reproduce (voz sintética) un mensaje.
5		Permite <ul style="list-style-type: none"> <li>• Continuar</li> </ul>
6	Selecciona "Continuar"	
7		Va a la <u>sección "terminación"</u> de este caso de uso
8		El caso de uso termina
Flujos alternos 3.1 Valida la respuesta. Es incorrecta.		
	Actor	Sistema
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
Flujos alternos 2.1 Selecciona "No"		

2.1. 1		Valida la respuesta. Es correcta
2.1. 2		Muestra y reproduce (voz sintética) un mensaje.
2.1. 3		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>
2.1. 4	Selecciona "Continuar"	
2.1. 5		Va al paso 7 del flujo básico
<b>Flujos alternos 3.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.1 El actor selecciona la opción "Terminar".</b>		
2.1.1		Muestra el mensaje ¿Desea abandonar?" y permite: <ul style="list-style-type: none"> <li>Si</li> <li>No</li> </ul>
2.1.2	Selecciona "Si"	
2.1.3		Si viene del <u>CU Familiarización y análisis</u> Va al <u>CU Familiarización y análisis</u> Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u>

		Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u>

Tabla 12 Sección terminación

Flujo básico: Terminación		
	Actor	Sistema
1		Muestra la pregunta ¿En qué termina la palabra? Y Brinda la posibilidad de seleccionar de responder: <ul style="list-style-type: none"> <li>• n</li> <li>• s</li> <li>• vocal</li> <li>• Consonante diferente de n o s</li> </ul> Y permite <ul style="list-style-type: none"> <li>• Terminar</li> </ul>
2	Selecciona "n"	
3		Valida la respuesta. Es correcta
4		Muestra y reproduce (voz sintética) un mensaje.
5		Permite <ul style="list-style-type: none"> <li>• Continuar</li> </ul>
6	Selecciona "Continuar"	
7		Va a la <u>sección "terminación"</u> de este caso de uso
8		El caso de uso termina
Flujos alternos 3.1 Valida la respuesta. Es incorrecta.		

	Actor	Sistema
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.1 Selecciona “s”</b>		
2.1.1		Valida la respuesta. Es correcta
2.1.2		Muestra y reproduce (voz sintética) un mensaje.
2.1.3		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>
2.1.4	Selecciona “Continuar”	
2.1.5		Va al paso 7 del flujo básico
<b>Flujos alternos 3.1 Valida la respuesta. Es incorrecta.</b>		
	Actor	Sistema
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.2 Selecciona “vocal”</b>		
2.2.1		Valida la respuesta. Es correcta
2.2.2		Muestra y reproduce (voz sintética) un mensaje.
2.2.3		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>
2.2.4	Selecciona “Continuar”	

2.2.5		Va al paso 7 del flujo básico
<b>Flujos alternos 2.2.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
2.2.1.1		Muestra y reproduce (voz sintética) un mensaje.
2.2.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.3 Selecciona “Constante diferente de n o s”</b>		
2.3.1		Valida la respuesta. Es correcta
2.3.2		Muestra y reproduce (voz sintética) un mensaje.
2.3.3		Permite <ul style="list-style-type: none"> <li>• Continuar</li> </ul>
2.3.4	Selecciona “Continuar”	
2.3.5		Va al paso 7 del flujo básico
<b>Flujos alternos 2.3.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
2.3.2		Muestra y reproduce (voz sintética) un mensaje.
2.3.3		Va al paso 1 del flujo básico
<b>Flujos alternos 2.1 El actor selecciona la opción “Terminar”.</b>		
2.1.1		Muestra el mensaje ¿Desea abandonar?” y permite: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>

2.1.2	Selecciona "Si"	
2.1.3		<p>Si viene del <u>CU Familiarización y análisis</u></p> <p>Va al <u>CU Familiarización y análisis</u></p> <p>Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u></p> <p>Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u></p>

Sección resumen

Tabla 13 Sección resumen

Flujo básico: Terminación		
	Actor	Sistema
1		<p>Muestra información sobre la palabra y la pregunta ¿Lleva tilde la palabra?</p> <p>Y brinda las respuestas</p> <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> <li>• No sé</li> </ul> <p>Y permite</p> <ul style="list-style-type: none"> <li>• Terminar</li> </ul>
2	Selecciona "si"	
3		Valida la respuesta. Es correcta

4		Muestra y reproduce (voz sintética) un mensaje.
5		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>
6	Selecciona "Continuar"	
7		Si viene del <u>CU Familiarización y análisis</u> Va al <u>CU Familiarización y análisis</u> Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u> Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u>
8		El caso de uso termina
<b>Flujos alternos 3.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.1 Selecciona "No"</b>		
2.1.1		Valida la respuesta. Es correcta
2.1.2		Muestra y reproduce (voz sintética) un mensaje.
2.1.3		Permite <ul style="list-style-type: none"> <li>Continuar</li> </ul>

2.1. 4	Selecciona "Continuar"	
2.1. 5		<p>Va al <u>CU Familiarización y análisis</u></p> <p>Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u></p> <p>Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia de uso</u></p>
<b>Flujos alternos 3.1 Valida la respuesta. Es incorrecta.</b>		
	<b>Actor</b>	<b>Sistema</b>
3.1.1		Muestra y reproduce (voz sintética) un mensaje.
3.1.2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.2 Selecciona "No sé"</b>		
2.2. 1		Muestra información sobre el tema
2.2. 2		Va al paso 1 del flujo básico
<b>Flujos alternos 2.1 El actor selecciona la opción "Terminar".</b>		
2.1.1		<p>Muestra el mensaje ¿Desea abandonar?" y permite:</p> <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
2.1.2	Selecciona "Si"	
2.1.3		<p>Si viene del <u>CU Familiarización y análisis</u></p> <p>Va al <u>CU Familiarización y análisis</u></p>



	<p>Si viene del <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u></p> <p>Va al <u>CU Realizar entrenamiento del tipo Dictado e inferencia</u></p>
--	--

**Tabla 14 Descripción del CU Realizar entrenamiento d tipo "Escritura con apoyo"**

<b>Objetivo</b>	Realizar los ejercicio como parte del entrenamiento	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario selecciona comenzar. El entrenamiento para la modalidad significado-palabra donde se dictan oraciones o frases que deberán ser escritas correctamente por el usuario.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Debe haberse seleccionado el tipo de entrenamiento "Escritura libre" de la modalidad "oración-frase" con una forma de presentación.	
<b>Postcondiciones</b>	Se realizan o no, el o los ejercicios. Se guardan los datos en la traza del usuario	
<b>Flujo de eventos</b>		
<b>Flujo básico: Realizar entrenamiento de tipo "Escritura con apoyo"</b>		
	<b>Actor</b>	<b>Sistema</b>
<b>16.</b>	El caso de uso se inicia cuando el actor selecciona la opción "Comenzar".	
<b>17.</b>		<p>Se muestran en la pantalla la frase u oración con espacios en blancos para completarla así como los datos:</p> <ul style="list-style-type: none"> <li>• Certamen.</li> </ul>

		<ul style="list-style-type: none"> <li>• Total.</li> <li>• Puntos.</li> <li>• Tiempo.</li> <li>• Autor de la frase.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Enviar</li> <li>• Ver frase</li> <li>• Reglas</li> <li>• Ayuda</li> <li>• Escuchar</li> <li>• Regresar</li> </ul>
18.	Escribe en el campo las palabras y signos de puntuación.	
19.	Selecciona la opción "Enviar"	
20.		Muestra un mensaje de información diciendo si es correcta o no la respuesta.
21.		Si los ejercicios no se han acabado vuelve al Paso 2 del flujo básico.
22.		
<b>Flujos alternos</b>		
<b>Flujos alternos 4.4 El actor selecciona la opción "Ver frase".</b>		
	<b>Actor</b>	<b>Sistema</b>
4.4.1		Ver <u>CU Mostrar opciones del ejercicio.</u> <u>Sección Ver frase</u>
4.4.2		El caso de uso termina
<b>Flujos alternos 4.5 El actor selecciona la opción "Reglas".</b>		

4.5.1		Ver <u>CU Mostrar opciones del ejercicio. Sección Reglas</u>
4.5.2		El caso de uso termina
<b>Flujos alternos 4.6 El actor selecciona la opción “Escuchar”.</b>		
4.6.1		Ver <u>CU Mostrar opciones del ejercicio. Sección Escuchar</u>
4.6.2		El caso de uso termina
<b>Flujos alternos 4.7 El actor selecciona la opción “Ayuda”.</b>		
4.7.1		Ver <u>CU Mostrar opciones del ejercicio. Sección Ayuda</u>
4.7.2		El caso de uso termina
<b>Flujos alternos 4.8 El actor selecciona la opción “Regresar”.</b>		
4.8.1		Muestra el mensaje “Entrenamiento no concluido ¿Desea abandonar?” y permite: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
4.8.2	Selecciona “Si”	
4.8.3		Va al <u>CU Listar modalidades de entrenamiento</u>
4.8.4		El caso de uso termina
<b>Flujos alternos 4.8.2 El actor selecciona la opción “No”.</b>		
4.8.2.1		Vuelve al paso 2 del flujo básico.
<b>Flujos alternos 6.2 Se acaban los ejercicios</b>		
6.2.1		Va al <u>CU Consultar trazas. Sección 1</u>

		El caso de uso termina.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Mostrar opciones de ejercicios.

**Tabla 15 Descripción del CU Realizar entrenamiento de tipo Escritura libre**

<b>Objetivo</b>	Realizar los ejercicio como parte del entrenamiento	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario selecciona comenzar. Entrenamiento para la modalidad significado-palabra donde se dictan oraciones o frases que deberán ser escritas correctamente por el usuario. Se aumentan 2 puntos por cada palabra correcta y se resta 1 por cada palabra incorrecta al igual que si se selecciona “pista” o “ver palabra”.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Debe haberse seleccionado el tipo de entrenamiento “Escritura libre” de la modalidad “oración-frase” con una forma de presentación.	
<b>Postcondiciones</b>	Se realizan o no, el o los ejercicios. Se guardan los datos en la traza del usuario	
<b>Flujo de eventos</b>		
<b>Flujo básico: Realizar entrenamiento de tipo “Dictado e inferencia”</b>		
	<b>Actor</b>	<b>Sistema</b>
<b>23.</b>	El caso de uso se inicia cuando el actor selecciona la opción “Comenzar”.	
<b>24.</b>		Lo de hablar  Brinda la posibilidad de escribir en un campo de texto ,muestra los datos:

		<ul style="list-style-type: none"> <li>• Certamen.</li> <li>• Total.</li> <li>• Puntos.</li> <li>• Tiempo.</li> <li>• Significado.</li> <li>• Palabra escrita</li> <li>• Palabra correcta</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Comprobar</li> <li>• Ver palabra</li> <li>• Pista</li> <li>• Reglas</li> <li>• Ayuda</li> <li>• Escuchar</li> </ul>
25.	Escribe la palabra en el campo	
26.	Selecciona la opción "Comprobar"	
27.		Valida los datos
28.		Si los ejercicios no se han acabado vuelve al Paso 2 del flujo básico.
<b>Flujos alternos 5.1 Los datos son incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
5.1.1		Muestra la palabra escrita por el usuario (incorrecta) y la palabra escrita de forma correcta.
4.4.2		Va al paso 2 del flujo básico.
<b>Flujos alternos</b>		

Flujos alternos 4.4 El actor selecciona la opción "Ver palabra".		
	Actor	Sistema
4.4.1		Ver <u>CU Mostrar opciones del ejercicio.</u> <u>Sección Ver palabra</u>
4.4.2		El caso de uso termina
Flujos alternos 4.5 El actor selecciona la opción "Reglas".		
4.5.1		Ver <u>CU Mostrar opciones del ejercicio.</u> <u>Sección Reglas</u>
4.5.2		El caso de uso termina
Flujos alternos 4.6 El actor selecciona la opción "Escuchar".		
4.6.1		Ver <u>CU Mostrar opciones del ejercicio.</u> <u>Sección Escuchar</u>
4.6.2		El caso de uso termina
Flujos alternos 4.7 El actor selecciona la opción "Ayuda".		
4.7.1		Ver <u>CU Mostrar opciones del ejercicio.</u> <u>Sección Ayuda</u>
4.7.2		El caso de uso termina
Flujos alternos 4.8 El actor selecciona la opción "Regresar".		
4.8.1		Muestra el mensaje "Entrenamiento no concluido ¿Desea abandonar?" y permite: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
4.8.2	Selecciona "Si"	
4.8.3		Va al <u>CU Listar modalidades de entrenamiento</u>

4.8.4		El caso de uso termina
<b>Flujos alternos 4.8.2 El actor selecciona la opción “No”.</b>		
4.8.2.1		Vuelve al paso 2 del flujo básico.
<b>Flujos alternos 6.2 Se acaban los ejercicios</b>		
6.2.1		Va al <u>CU Consultar trazas. Sección 1</u>
6.2.2		El caso de uso termina.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Mostrar opciones de ejercicios.

**Tabla 16. Descripción del CU Mostrar opciones de ejercicios.**

<b>Objetivo</b>	Realizar diferentes funciones de ayuda para la resolución de los ejercicios del entrenador.
<b>Actores</b>	Usuario
<b>Resumen</b>	Diversas opciones que se presentan en los ejercicios de entrenamiento para apoyar y ayudar al usuario. Las opciones se muestran de acuerdo al tipo de ejercicio,
<b>Complejidad</b>	Baja
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	Debe haberse seleccionado algún tipo de opción: ver pista, ver palabra, escuchar palabra, ver frase, reglas, ayuda.
<b>Postcondiciones</b>	Se realizan o no, el o los ejercicios. Se guardan los datos en la traza del usuario
<b>Flujo de eventos</b>	
<b>Flujo básico: Opciones de ejercicios</b>	

	Actor	Sistema
29.	El actor selecciona pista	
30.		Muestra la palabra ocultando los posibles accidentes gramaticales de esta.
31.		El caso de uso termina

## Flujo básico: Ver palabra

	Actor	Sistema
1	El actor selecciona "Ver palabra"	
2		Muestra la palabra de forma intermitente durante unos segundos
3		El caso de uso termina

## Flujo básico: Escuchar palabra

	Actor	Sistema
32.	El actor selecciona "Escuchar palabra"	
33.		Mediante voz sintética, reproduce la palabra o frase en cuestión.
34.		El caso de uso termina

## Flujo básico: Reglas

	Actor	Sistema
35.	El actor selecciona "Reglas"	



36.		Muestra las ortográficas
37.		El caso de uso termina

Flujo básico: Opciones de ejercicios		
	Actor	Sistema
38.	El actor selecciona "Ayuda"	
39.		Muestra la ayuda del ejercicio
40.		El caso de uso termina

Tabla 17. Descripción del CU Realizar entrenamiento de tipo "Familiarización y análisis".

<b>Objetivo</b>	Familiarizarse con las palabras seleccionadas.	
<b>Actores</b>	Usuario	
<b>Resumen</b>	Entrenamiento para la modalidad significado-palabra donde se presentan listados de palabras para que el usuario interactúe con ellas, escuche su pronunciación, lea su significado y pueda realizar un análisis de la palabra.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Debe haberse seleccionado el tipo de entrenamiento "Familiarización y análisis" de la modalidad "significado palabra" con una forma de presentación.	
<b>Postcondiciones</b>	Se realizan o no, el o los ejercicios. Se guardan los datos en la traza del usuario	
<b>Flujo de eventos</b>		
<b>Flujo básico: Realizar entrenamiento de tipo "Escritura con apoyo"</b>		
	<b>Actor</b>	<b>Sistema</b>

<b>Objetivo</b>	Registrar usuario en el sistema	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El usuario, para poder comenzar a utilizar la aplicación, ha de introducir su nombre, apellido y seleccionar el tipo de voz (voz sintética con la que el sistema interactuará con el usuario en lo adelante), como parte de su registro en el sistema. Estos datos son guardados junto a la fecha y hora actual como parte del identificador de una traza que recogerá las estadísticas de todos los ejercicios que realice.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Debe haberse iniciado la aplicación	
<b>Postcondiciones</b>	Se registra el usuario en el sistema	
<b>Flujo de eventos</b>		
<b>Flujo básico: Registrar usuario en el sistema</b>		
	<b>Actor</b>	<b>Sistema</b>
41.	El caso de uso se inicia cuando el usuario inicia la aplicación	
42.		Brinda la posibilidad de introducir datos en los campos nombre, apellidos y seleccionar una opción de voz (Forma en que el sistema interactúa con el usuario mediante la síntesis de voz) en seleccionar voz.  Y permite: <ul style="list-style-type: none"> <li>• Registrarse</li> <li>• Salir</li> </ul>
43.	Introduce los datos nombre y apellidos	
44.	Selecciona la voz "Jorge"	

45.		Presenta al usuario, dándole una bienvenida.
46.	Selecciona la opción "Registrarse"	
47.		Valida los datos nombre, apellido y voz (selecciona voz).
48.		Registra al usuario
49.		Va al <u>CU Listar modalidades de entrenamiento</u>
50.		Termina el caso de uso
<b>Flujos alternos</b>		
<b>7.1 Los datos están incorrectos.</b>		
	<b>Actor</b>	<b>Sistema</b>
4.1.1		Muestra un mensaje informando que los datos están incorrectos
4.1.2		<b>Va al paso 2 del flujo básico</b>
<b>Flujos alternos</b>		
<b>4.1 El usuario selecciona la opción "Salir"</b>		
	<b>Actor</b>	<b>Sistema</b>
4.1.1		Muestra el mensaje "Desea salir del entrenador" y permite las opciones: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
4.1.1	Selecciona Si	
4.1.2		Se cierra la aplicación
		Termina el caso de uso

Flujos alternos		
4.1.1.2 Selecciona No		
	Actor	Sistema
4.1.1		Va al paso 2 del flujo básico
Relaciones	CU Incluidos	
	CU Extendidos	
51.	Click en el botón "Comenzar".	
52.		Muestra un listado de palabras y permite <ul style="list-style-type: none"> <li>• Seleccionar palabra</li> <li>• Regresar</li> <li>• Reglas</li> <li>• Escuchar</li> <li>• Próxima</li> <li>• Anterior</li> <li>• Realizar análisis</li> <li>• Ayuda</li> </ul>
53.	Selecciona una palabra	
54.		Dice la palabra(la palabra se escucha)  Muestra el significado de la palabra
55.	Selecciona Realizar análisis	
56.		Va al <i>CU Realizar análisis de palabra</i>
57.		El caso de uso termina
Flujos alternos		

Flujos alternos 3.1 El actor selecciona la opción “Próxima”.		
	Actor	Sistema
3.4.1		Si hay más páginas, pasa a la siguiente
3.4.2		Va al paso 2 del flujo básico
Flujos alternos 3.1 El actor selecciona la opción “Anterior”.		
	Actor	Sistema
3.4.1		Si hay más páginas, pasa a la anterior
3.4.2		Va al paso 2 del flujo básico
Flujos alternos 3.5 El actor selecciona la opción “Reglas”.		
3.5.1		Ver <u>CU Mostrar opciones del ejercicio. Sección Reglas</u>
3.5.2		Vuelve al paso 2 del flujo básico
Flujos alternos 3.6 El actor selecciona la opción “Escuchar”.		
3.6.1		Ver <u>CU Mostrar opciones del ejercicio. Sección Escuchar</u>
3.6.2		Vuelve al paso 2 del flujo básico
Flujos alternos 3.7 El actor selecciona la opción “Ayuda”.		
3.7.1		Ver <u>CU Mostrar opciones del ejercicio. Sección Ayuda</u>
3.7.2		Vuelve al paso 2 del flujo básico
Flujos alternos 4.8 El actor selecciona la opción “Regresar”.		
3.8.1		Muestra el mensaje “¿Desea abandonar?” y permite: <ul style="list-style-type: none"> <li>• Si</li> </ul>

		<ul style="list-style-type: none"> <li>No</li> </ul>
3.8.2	Selecciona "Si"	
3.8.3		Va al <u>CU Listar opciones de entrenamiento</u>
3.8.4		El caso de uso termina
<b>Flujos alternos 4.8.2 El actor selecciona la opción "No".</b>		
3.8.2.1		Vuelve al paso 2 del flujo básico.
<b>Flujos alternos 3.9 El actor selecciona "Analizar palabra"</b>		
3.9.1		Va al <u>CU Analizar palabra</u>
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	<p>Mostrar opciones de ejercicios.</p> <p>Analizar palabra</p>

<b>Flujo básico: Selecciona ejercicios de la modalidad Significado-Palabra</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona ejercicios de la modalidad Significado-Palabra	
2		<p>Brinda la posibilidad de seleccionar los niveles</p> <ul style="list-style-type: none"> <li>Escritura con apoyo</li> <li>Escritura libre</li> </ul> <p>Y la forma de presentación con campos para introducir números:</p> <ul style="list-style-type: none"> <li>Secuencial</li> <li>Aleatoria</li> </ul>

3	Selecciona Escritura con apoyo	
4	Selecciona secuencial	
5	Introduce los datos	.
6		Valida los datos
7	Selecciona Comenzar	
8		Va al <u>CU Realizar ejercicios del tipo Escritura con apoyo</u>
9		El caso de uso termina

### Flujos alternos 3.1 El actor selecciona Escritura libre

	Actor	Sistema
3.1.1	Selecciona secuencial	
5.1.2	Introduce los datos	
5.1.3		Valida los datos
5.1.4	Selecciona comenzar	
5.1.5		Va al <u>CU Realizar ejercicios del Escritura libre</u>
		El caso de uso termina

### Flujos alternos 3.1.1.1 El actor selecciona aleatorio

	Actor	Sistema
3.1.1. 2		Valida los datos
3.1.1. 3		Va al <u>CU Realizar ejercicios del Escritura libre</u>
3.1.1. 4		El caso de uso termina

Flujos alternos 4.2 El actor selecciona aleatorio		
	Actor	Sistema
4.2.1		Valida los datos
4.2.2		Va al CU <u>Realizar ejercicios del tipo Escritura con apoyo</u>
4.2.3		El caso de uso termina

Tabla 18 Descripción del CU Configurar modalidades de entrenamiento

<b>Objetivo</b>	Configurar las opciones para realizar los entrenamientos	
<b>Actores</b>	Usuario	
<b>Resumen</b>	Se listan las bases de datos disponibles para realizar los entrenamientos agrupándolos en dos modalidades: Significado-Palabra, contienen diversas palabras (agrupadas según el nivel de dificultad o temática) con su respectivo significado; Oración-Frase, con disimiles frases u oraciones agrupadas por materias. Muestra la forma en que desea se presenten los ejercicios de entrenamiento: Secuencial o al azar; además del tipo de entrenamiento que se quiere realizar de acuerdo a la modalidad que se seleccione.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	De be haberse registrado en el sistema	
<b>Postcondiciones</b>	Se va a un tipo de entrenamiento	
<b>Flujo de eventos</b>		
<b>Flujo básico: Listar opciones de ejercicios</b>		
	<b>Actor</b>	<b>Sistema</b>



Flujo básico: Seleccionar acción		
	Actor	Sistema
1	El caso de uso se inicia cuando termina uno de los tipos de ejercicio "Dictado e inferencia", "Escritura con apoyo", "Escritura libre"	
2		Brinda la posibilidad de <ul style="list-style-type: none"> <li>• Seguir entrenando</li> <li>• Consultar trazas</li> <li>• Salir</li> </ul>
3	Selecciona consultar trazas	
4		Va al flujo básico del <u>CU Consultar Traza</u>
5		El caso de uso termina
Flujos alternos 3.1 El actor selecciona Seguir entrenando.		
	Actor	Sistema
3.1.1		Va al <u>CU Listar Opciones de entrenamiento</u>
3.1.2		El caso de uso termina
Flujos alternos 3.2 El actor selecciona Salir.		
	Actor	Sistema
3.2.1		El caso de uso termina

58.	El caso de uso se inicia cuando el actor se registra	
59.		<p>Brinda la posibilidad de</p> <ul style="list-style-type: none"> <li>• Seleccionar ejercicios de la modalidad Oración-Frase</li> <li>• Sección 1: Seleccionar ejercicios de la modalidad Significado-Palabra</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Regresar</li> <li>• Ver trazas</li> </ul>
60.	Selecciona ejercicios de la modalidad Oración-Frase	
61.		<p>Brinda la posibilidad de seleccionar los niveles</p> <ul style="list-style-type: none"> <li>• Familiarización y análisis</li> <li>• Dictado e inferencia</li> </ul> <p>Y la forma de presentación con campos para introducir números:</p> <ul style="list-style-type: none"> <li>• Secuencial</li> <li>• Aleatoria</li> </ul>
62.	Selecciona Familiarización y análisis	
63.	Selecciona secuencial	
64.	Introduce los datos	.
65.		Valida los datos

66.	Selecciona Comenzar	
67.		Va al <u>CU Realizar ejercicios del tipo Familiarización y análisis</u>
68.		El caso de uso termina
<b>Flujos alternos 5.1 El actor selecciona Dictado e inferencia</b>		
	<b>Actor</b>	<b>Sistema</b>
5.1.1	Selecciona secuencial	
5.1.2	Introduce los datos	
5.1.3		Valida los datos
5.1.4	Selecciona comenzar	
5.1.5		Va al CU <u>Realizar ejercicios del tipo Dictado e inferencia</u>
		El caso de uso termina
<b>Flujos alternos 5.1.1.1 El actor selecciona aleatorio</b>		
	<b>Actor</b>	<b>Sistema</b>
5.1.1. 2		Valida los datos
5.1.1. 3		Va al CU <u>Realizar ejercicios del tipo Dictado e inferencia</u>
5.1.1. 4		El caso de uso termina
<b>Flujos alternos 5.2 El actor selecciona aleatorio</b>		
	<b>Actor</b>	<b>Sistema</b>
5.2.1		Valida los datos

5.2.2		Va al CU <u>Realizar ejercicios del tipo Dictado e inferencia</u>
5.2.3		El caso de uso termina
<b>Flujos alternos 3.1 El actor selecciona la opción “Ver trazas”.</b>		
3.1.1		Va al <u>CU Consultar trazas</u>
3.1.2		El caso de uso termina
<b>Flujos alternos 3.2 El actor selecciona la opción “Regresar”.</b>		
3.2.1		Va al <u>CU Registrar usuario en el sistema</u>
3.2.2		El caso de uso termina
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	Mostrar opciones de ejercicios.

Anexo 3 Modelos Conceptuales de Comunicación (MCC)

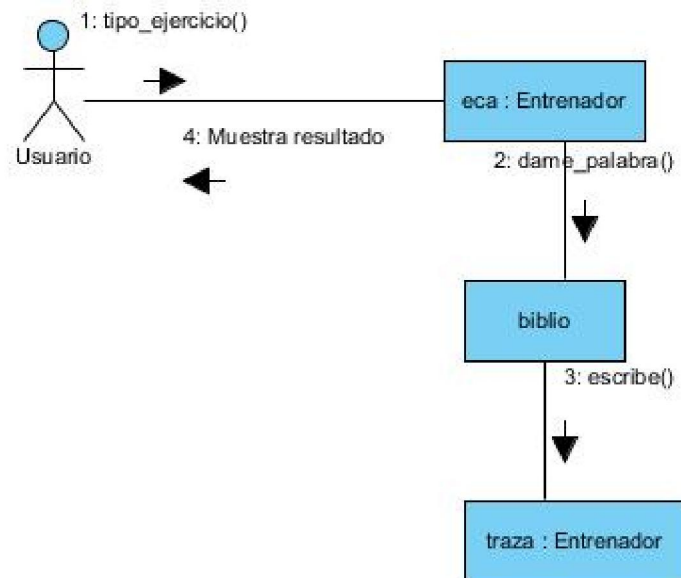


Figura 16 MCC del CU Realizar ejercicio de tipo "Escritura con apoyo"

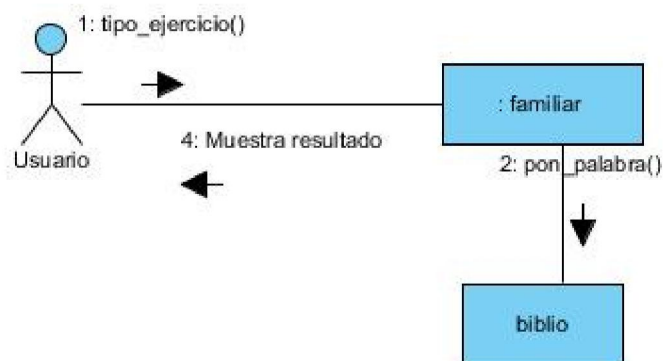


Figura 17 MCC del CU Realizar ejercicio de tipo "Familiarización y análisis"

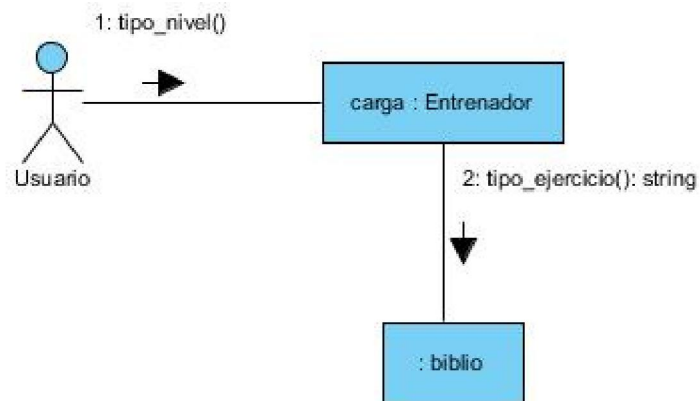


Figura 18 MCC del CU Configurar Modalidades de entrenamiento

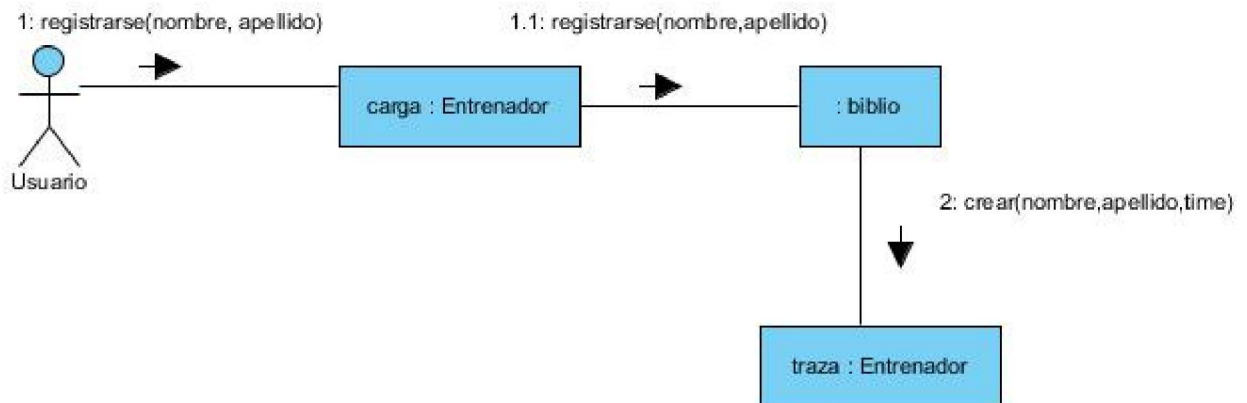


Figura 19 MCC del CU Registrarse

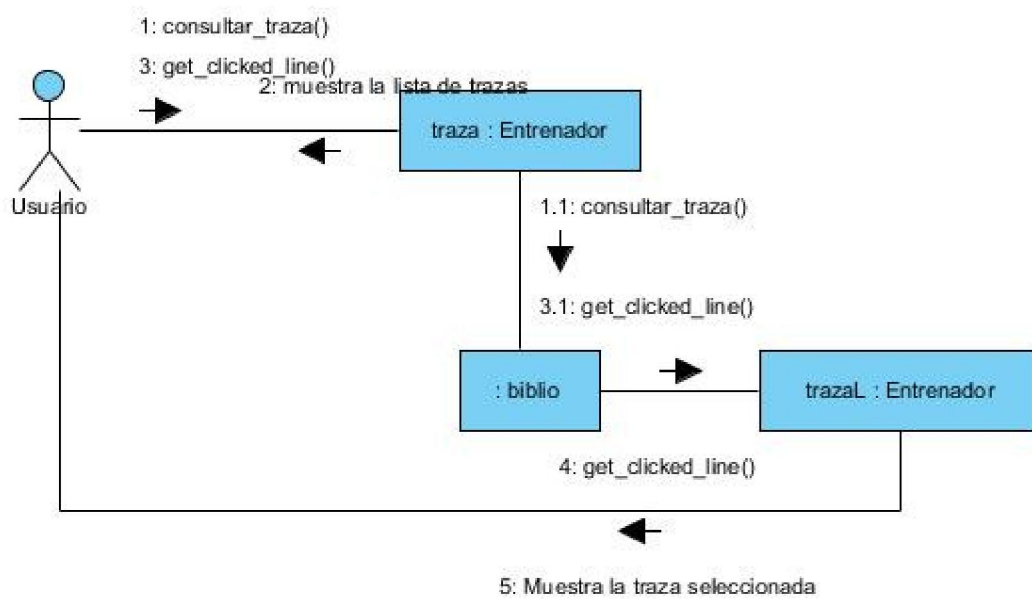


Figura 20 MCC del CU Consultar traza

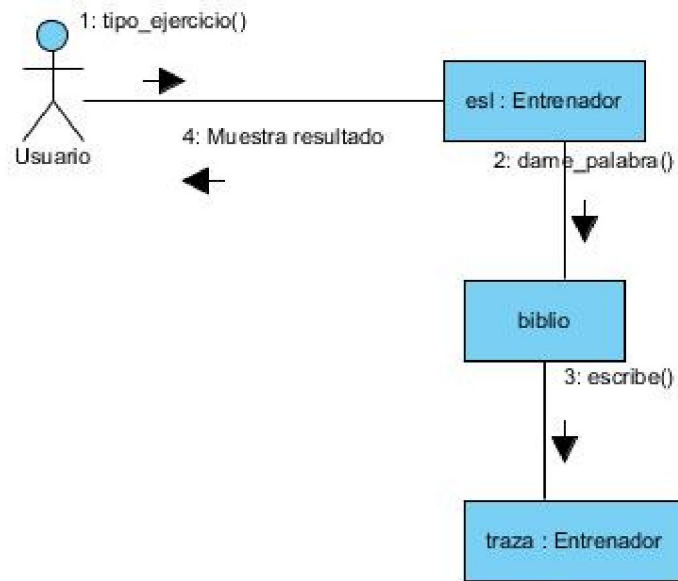


Figura 21 MCC del CU Realizar ejercicio de tipo "Escritura libre"

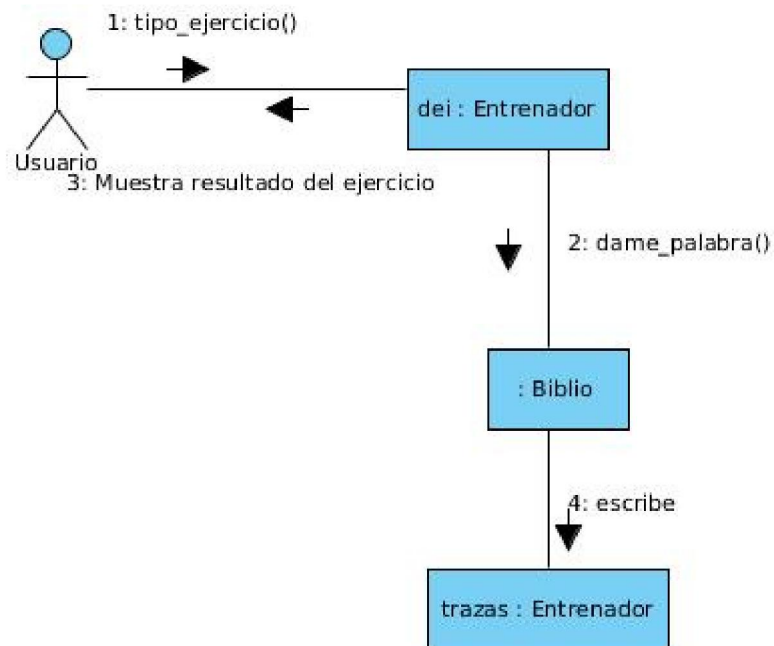


Figura 22 MCC del CU Realizar ejercicio de tipo "Dictado e inferencia"

**Anexo 4 Descripción de casos de prueba**

**CP Registrar usuario en el sistema**

Tabla 19. CP Registrar usuario en el sistema.

Escenario	Descripción	Nombre	Apellidos	Voz (Seleccione voz)	Respuesta del sistema	Flujo central
EC 1.1	El caso de uso se inicia cuando el usuario inicia la aplicación				Se muestran en la pantalla los campos nombre, apellidos y la opción seleccionar voz. Y permite: • Registrarse • Salir	Iniciar aplicación
EC 1.2	Introduce los datos nombre y apellidos	V	V	N/A		Iniciar aplicación
EC 1.3	Selecciona la voz "Jorge"	V	N/A	V	Recibe, hablándole al usuario con la voz de Jorge.	Iniciar aplicación
EC 1.4	Selecciona la opción "Registrarse"	V	V	V	Verifica los datos. Termina el caso de prueba	Iniciar aplicación/Registrar
EC 1.5	Los datos están incorrectos	I I I I V V	I I V V I I	I V I V I V	Muestra un mensaje informando que los datos están incorrectos	Iniciar aplicación/Registrar



		V	V	I		
		V	V	V		
EC 1.6	El usuario selecciona la opción salir	N/A	N/A	N/A		

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre	Cadena	NO	Cadena de caracteres donde la primera letra ha de ser mayúscula. No se permiten números
2	apellido	Cadena	NO	Cadena de caracteres donde la primera letra ha de ser mayúscula. No se permiten números
3	Tipo de voz	Cadena	NO	Jorge, Diego, Esperanza

## CP Listar modalidades de entrenamiento

Tabla 20. CP Configurar modalidades de entrenamiento.

SC <Oración - frase>

Escenario	Descripción	Modalidad	Nivel	Presentación	Respuesta del sistema	Flujo central
EC 1.1	El caso de uso se inicia cuando el actor se registra	N/A	N/A N/A	N/A	Brinda la posibilidad de • Seleccionar	Iniciar aplicación/Registrarse/

					<p>ejercicios de la modalidad Oración-Frase</p> <ul style="list-style-type: none"> <li>• Sección 1: Seleccionar ejercicios de la modalidad Significado-Palabra</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Regresar</li> <li>• Ver trazas</li> </ul>	
EC 1.2	<p>Selecciona Familiarización y análisis</p> <p>Selecciona secuencial</p> <p>Introduce los datos</p>	V	V	V	Valida los datos	Iniciar aplicación/Registrarse/
EC 1.3	<p>Selecciona Comenzar</p>	V	N/A	V	<p>Va al <u>CP Realizar ejercicios del tipo Familiarización y análisis</u></p>	<p>Iniciar aplicación/Registrarse/</p> <p>Iniciar aplicación/Registrarse/</p>
EC 1.4	<p>El actor selecciona Dictado e inferencia.</p> <p>Selecciona secuencial</p> <p>Introduce los datos</p>	V	V	V	<p>Va al CU Realizar ejercicios del tipo Dictado e inferencia</p>	Iniciar aplicación/Registrar
EC 1.5	<p>El actor selecciona aleatorio</p>	V	V	V	<p>Valida los datos</p> <p>Va al CU Realizar ejercicios del tipo</p>	Iniciar aplicación/Registrar

					Dictado e inferencia	
EC 1.6	El usuario selecciona la opción Regresar	N/A	N/A	N/A	Va al CP Registrar usuario en el sistema	Iniciar aplicación/Registrar

## SC <Significado - palabra>

Escenario	Descripción	Modalidad	Nivel	Presentación	Respuesta del sistema	Flujo central
EC 1.1	Selecciona ejercicios de la modalidad Significado-Palabra	N/A	N/A	N/A	Brinda la posibilidad de seleccionar los niveles <ul style="list-style-type: none"> <li>• Escritura con apoyo</li> <li>• Escritura libre</li> </ul> Y la forma de presentación con campos para introducir números: <ul style="list-style-type: none"> <li>• Secuencial</li> <li>• Aleatoria</li> </ul>	Iniciar aplicación/Registrarse/
EC 1.2	Selecciona Escritura con apoyo Selecciona	V	V	V	Valida los datos	Iniciar aplicación/Registrarse/

	secuencial Introduce los datos					
EC 1.3	Selecciona Comenzar	V	N/A	V	Va al CP Realizar ejercicios del tipo Escritura con apoyo	Iniciar aplicación/Registrarse/ Iniciar aplicación/Registrarse/
EC 1.4	El actor selecciona Escritura libre. Selecciona secuencial Introduce los datos	V	V	V	Va al CU Realizar ejercicios del tipo Dictado e inferencia	Iniciar aplicación/Registrar
EC 1.5	El actor selecciona aleatorio	V	V	V	Valida los datos Va al CP Realizar ejercicios del tipo Escritura libre	Iniciar aplicación/Registrar
EC 1.6	El usuario selecciona la opción Regresar	N/A	N/A	N/A	Va al CP Registrar usuario en el sistema	

**CP Realizar ejercicio  
de tipo dictado e  
inferencia**

Escenario	Descripción	Campo palabra	Respuesta del sistema	Flujo central
EC 1.1	El actor selecciona la opción "Comenzar".		Se muestran en la pantalla los campos nombre, apellidos y la opción seleccionar voz. Y permite: • Registrarse • Salir	Iniciar aplicación
EC 1.2	Introduce la palabra en el campo	V		Iniciar aplicación
EC 1.3	Selecciona la opción "Comprobar"	V	Se validan los datos.	Iniciar aplicación
EC 1.4	Los datos son incorrectos"	I	Verifica los datos. Termina el caso de prueba	Iniciar aplicación/Registrar
EC 1.5	El actor selecciona la opción "Ver palabra".	N/A	<u><i>Ver CP Mostrar opciones del ejercicio. Sección Ver palabra</i></u>	Iniciar aplicación
EC 1.6	El actor selecciona la opción "Reglas".	N/A	<u><i>Ver CP Mostrar opciones del ejercicio. Sección Reglas</i></u>	Iniciar aplicación
EC 1.7		N/A		Iniciar aplicación.

	El actor selecciona la opción "Escuchar".		<u>Ver CP Mostrar opciones del ejercicio. Sección Escuchar</u>	
EC 1.8	El actor selecciona la opción "Ayuda".	N/A	<u>Ver CP Mostrar opciones del ejercicio. Sección Ayuda.</u>	Iniciar aplicación.
EC 1.9	El actor selecciona la opción "Regresar".	N/A	Muestra el mensaje "Entrenamiento no concluido ¿Desea abandonar?" y permite: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>	Iniciar aplicación.
EC 1.10	El actor selecciona Si	N/A	Va al CP Listar modalidades de entrenamiento	Iniciar aplicación/Cerrar
EC 1.11	El actor selecciona la opción "No".	N/A	Vuelve al paso 2 del flujo básico. Va al <u>CP Consultar trazas. Sección 1</u> El caso de uso termina.	Iniciar aplicación/Cerrar

No	Nombre de campo	Clasificación	Descripción
1	Campo palabra	Cadena	Cadena de caracteres que conforman la palabra

## CP Mostrar opciones de ejercicios

Tabla 21. Mostrar opciones de ejercicios.

SC <Pista>

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1	El actor selecciona pista	Muestra la palabra ocultando los posibles accidentes gramaticales de esta.	Iniciar aplicación/Algún Tipo de ejercicio/Pista

## SC <Ver palabra>

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1	El actor selecciona "Ver palabra"	Muestra la palabra de forma intermitente durante unos segundos	Iniciar aplicación/Algún Tipo de ejercicio/Ver palabra

## SC <Reglas>

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1	El actor selecciona "Reglas"	Muestra la palabra de forma intermitente durante unos segundos	Iniciar aplicación/Algún Tipo de ejercicio/Reglas

## SC <Ayuda>

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1		Muestra la ayuda del ejercicio	Iniciar aplicación/Algún Tipo de ejercicio/Ayuda

	El actor selecciona "Ayuda"		
--	-----------------------------	--	--

Tabla 22. CP Realizar ejercicio de tipo Escritura con apoyo.

**CP Realizar ejercicio de tipo Escritura con apoyo**

Escenario	Descripción	Campo oración	Respuesta del sistema	Flujo central
EC 1.1	El actor selecciona la opción "Comenzar".		<p>Se muestran en la pantalla la frase u oración con espacios en blancos para completarla así como los datos:</p> <ul style="list-style-type: none"> <li>• Certamen.</li> <li>• Total.</li> <li>• Puntos.</li> <li>• Tiempo.</li> <li>• Autor de la frase.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Clic para tildes, mayúsculas y signos de puntuación</li> <li>• Escribir grafemas</li> <li>• Reiniciar</li> <li>• Edición libre</li> <li>• Enviar</li> <li>• Ver frase</li> <li>• Reglas</li> <li>• Ayuda</li> <li>• Escuchar</li> <li>• Regresar</li> </ul>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo



EC 1.2	Selecciona la opción "Escribir grafema"	V	Brinda la posibilidad de escribir los grafemas de las palabras incompletas.	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.3	Escribe los grafemas de las palabras incompletas. Selecciona la opción "Enviar"	V	Muestra un mensaje de información diciendo si es correcta o no la respuesta. Si los ejercicios no se han acabado vuelve al Paso 2 del flujo básico. De los contrario, el caso de uso termina	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.4	El actor selecciona la opción "Clic para tildes, mayúsculas y signos de puntuación"	N/A	Brinda la posibilidad de agregar tildes, signos de puntuación, quitar o poner mayúsculas. Va al paso 2 del flujo básico	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.5	El actor selecciona la opción "Reiniciar"	N/A	<u>Se reinicia el ejercicio</u> <u>Va al paso 2 del flujo básico</u>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.6	El actor selecciona la opción "Edición libre".	N/A	<u>Ver CP Mostrar opciones del ejercicio. Sección Reglas</u>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.7	El actor selecciona la opción "Escuchar".	N/A	<u>Permite redactar sin restricciones</u> <u>Va al paso 2 del flujo básico</u>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.8		N/A		

	El actor selecciona la opción "Ver frase"...		<u>Ver CP Mostrar opciones del ejercicio. Sección Ver frase</u> <u>El caso de uso termina</u>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.9	El actor selecciona la opción "Regresar".	N/A	Muestra el mensaje "Entrenamiento no concluido ¿Desea abandonar?" y permite: • Si • No .	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.10	El actor selecciona Si	N/A	Va al CP Listar modalidades de entrenamiento	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.11	El actor selecciona la opción "Reglas"	N/A	Ver CP Mostrar opciones del ejercicio. Sección Reglas	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
			El caso de uso termina	
EC 1.12	El actor selecciona la opción "Escuchar".	N/A	Ver CP Mostrar opciones del ejercicio. Sección Escuchar	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
			El caso de uso termina	
EC 1.13	El actor selecciona la opción "Ayuda".	N/A	Ver CU Mostrar opciones del ejercicio. Sección Ayuda	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
			El caso de uso termina	

No	Nombre de campo	Clasificación	Descripción
----	-----------------	---------------	-------------

1	Campo palabra	Cadena	Cadena de caracteres que conforman la palabra
---	---------------	--------	---

**CP Realizar ejercicio de tipo Escritura libre**

**Tabla 23. Realzar ejercicio de tipo Escritura Libre.**

Escenario	Descripción	Campo oración	Respuesta del sistema	Flujo central
EC 1.1	El actor selecciona la opción "Comenzar".		<p>Se muestran en la pantalla la frase u oración con espacios en blancos para completarla así como los datos:</p> <ul style="list-style-type: none"> <li>• Certamen.</li> <li>• Total.</li> <li>• Puntos.</li> <li>• Tiempo.</li> <li>• Autor de la frase.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Clic para tildes, mayúsculas y signos de puntuación</li> <li>• Escribir grafemas</li> <li>• Reiniciar</li> <li>• Edición libre</li> <li>• Enviar</li> <li>• Ver frase</li> <li>• Reglas</li> <li>• Ayuda</li> <li>• Escuchar</li> <li>• Regresar</li> </ul>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.2	Selecciona la opción	V		Iniciar aplicación/Modalidad

	“Escribir grafema”	I	Brinda la posibilidad de escribir los grafemas de las palabras incompletas.	de entrenamiento/Escritura con apoyo/“Escribir grafema”
EC 1.3	Escribe los grafemas de las palabras incompletas. Selecciona la opción “Enviar”	V	Se valida la respuesta. Muestra un mensaje de información diciendo si es correcta o no la respuesta.	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo
EC 1.4	El actor selecciona la opción “Clic para tildes, mayúsculas y signos de puntuación”	N/A	Brinda la posibilidad de agregar tildes, signos de puntuación, quitar o poner mayúsculas.	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/ “Clic para tildes, mayúsculas y signos de puntuación”
EC 1.5	El actor selecciona la opción “Reiniciar”	N/A	<i>Se reinicia el ejercicio</i>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/Reiniciar
EC 1.6	El actor selecciona la opción “Edición libre”.	N/A	<i>Ver CP Mostrar opciones del ejercicio. Sección Reglas</i>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/Edición libre
EC 1.7	El actor selecciona la opción “Escuchar”.	N/A	<i>Permite redactar sin restricciones</i>	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/Escuchar
EC 1.8	El actor selecciona la	N/A	<i>Ver CP Mostrar opciones del ejercicio. Sección Ver</i>	Iniciar aplicación/Modalidad de

	opción "Ver frase"...		<u>frase</u> <u>El caso de uso termina</u>	entrenamiento/Escritura con apoyo/Ver frase
EC 1.9	El actor selecciona la opción "Regresar".	N/A	Muestra el mensaje "Entrenamiento no concluido ¿Desea abandonar?" y permite: • Si • No .	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/Regresar
EC 1.10	El actor selecciona Si	N/A	Va al CP Listar modalidades de entrenamiento	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/reglas
EC 1.11	El actor selecciona la opción "Reglas"	N/A	Ver CP Mostrar opciones del ejercicio. Sección Reglas El caso de uso termina	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/escuchar
EC 1.12	El actor selecciona la opción "Ayuda".	N/A	Ver CP Mostrar opciones del ejercicio. Sección Ayuda El caso de uso termina	Iniciar aplicación/Modalidad de entrenamiento/Escritura con apoyo/ayuda

No	Nombre de campo	Clasificación	Descripción
1	Campo oración	Cadena	Cadena de caracteres que conforman la oración