

Sistema de reconocimiento de palabras para la EDU-CIAA

Autor

Gonzalo Ávila Alterach - Padrón 94.950

Fecha:

2do cuatrimestre 2016

Tabla de contenido

[Registro de cambios](#)

[Acta de Constitución del Proyecto](#)

[1. Propósito del proyecto](#)

[2. Alcance del proyecto](#)

[3. Supuestos del proyecto](#)

[4. Requerimientos](#)

[5. Entregables principales del proyecto](#)

[6. Desglose del trabajo en tareas](#)

[7. Diagrama de Gantt](#)

[8. Gestión de riesgos](#)

[9. Gestión de la calidad](#)

Registro de cambios

Revisión	Cambios realizados	Fecha
1.0	Creación del documento y firma del Acta de Constitución	31/08/2016
1.1	Completado del resto de los puntos del documento	21/09/2016
1.2	Correcciones de formato, cambios al diagrama de Gantt	24/09/2016

Acta de Constitución del Proyecto

Buenos Aires, 31 de agosto de 2016

Por medio de la presente se acuerda con el Sr. Gonzalo Ávila Alterach que su Trabajo Final del Seminario de Sistemas Embebidos se titulará “Sistema de reconocimiento de palabras para la EDU-CIAA”, consistirá esencialmente en el prototipo preliminar de un conjunto de procedimientos y técnicas que permitan integrar un sistema para reconocer palabras aisladas en español, en cualquier aplicación basada en la plataforma EDU-CIAA, y tendrá un presupuesto preliminar estimado de 100 hs de trabajo, con fecha de inicio miércoles 12 de octubre de 2016 y fecha de finalización jueves 1 de diciembre de 2016.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg
Profesor Adjunto del Seminario de Sistemas Embebidos

1. Propósito del proyecto

El propósito del proyecto es desarrollar una biblioteca que permita reconocer palabras aisladas, en español.

Además se pretende que el proyecto permita el aprendizaje de diversos conceptos y metodologías relacionadas con el desarrollo de sistemas embebidos, y así aprobar la materia.

Asimismo, también se espera que el proyecto sea de utilidad para cualquier persona que desee integrar un sistema de estas características en un diseño existente.

2. Alcance del proyecto

El proyecto consistirá en el desarrollo de la biblioteca que esté compuesta de los siguientes módulos:

1. Adquisición desde una entrada analógica y almacenado de datos en memoria
2. Detección de comienzo y fin de palabras
3. Reconocimiento de palabras aisladas

Además se realizarán distintos tests para verificar la correcta implementación de la misma, y comparar la eficiencia con otras implementaciones similares.

También se realizará una pequeña aplicación de prueba, que incluya a la biblioteca desarrollada y transmita los resultados del reconocimiento a una computadora, para validar al prototipo.

No se incluyen en el marco del proyecto los siguientes elementos:

- Diseño de circuito amplificador para señal de micrófono
- Diseño de circuito de filtro para señal de micrófono
- Pantalla u otro tipo de visualización donde se observen las palabras habladas
- Creación y entrenamiento estadístico de los modelos de habla

3. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se poseen los conocimientos o asesoría necesarios para llevarlo a cabo, habiendo cursado una materia relacionada con el reconocimiento del habla
- Se dispone de los modelos de habla para español
- No se integrará el sistema de reconocimiento en una aplicación de misión crítica o de la que dependan vidas o salud de personas
- Se dispone de una placa EDU-CIAA
- Se tiene un módulo de hardware formado por un micrófono y un amplificador

4. Requerimientos

1. Requerimientos asociados con **adquisición y almacenamiento**
 - 1.1. Tasa de muestreo: 16 ksps
 - 1.2. Bits de muestreo: 10 bits
 - 1.3. Tipo de almacenamiento: continuo, en *buffer* circular
 - 1.4. Duración de almacenamiento: 2 segundos
2. Requerimientos asociados con **detección de comienzo y fin de palabras**
 - 2.1. Algoritmo a usar: basado en la energía de una ventana pequeña
 - 2.2. Error menor al 10% con una entrada con relación señal-ruido de 25 dB
3. Requerimientos asociados con **reconocimiento de palabras aisladas**
 - 3.1. Reconocimiento estadístico, basado en *cadena oculta de Markov* y coeficientes *MFCC*
 - 3.2. Soporte de modelos del tipo “mezcla de Gaussianas” con matriz de covarianza diagonal y formato propietario del toolkit *HTK*
 - 3.3. Cantidad máxima de palabras a reconocer: al menos 10 palabras distintas
 - 3.4. Cantidad máxima de fonemas por palabra: al menos 10 fonemas por palabra
 - 3.5. Tasas de error dentro de un 5% de las obtenidas con el toolkit *HTK*
4. Requerimientos generales asociados con la **biblioteca**
 - 4.1. Posibilidad de integración a una aplicación existente
 - 4.2. Soporte para obtener las N palabras más probables y una cantidad que las relacione

5. Entregables principales del proyecto

- Código fuente (biblioteca para reconocimiento)
- Código fuente (aplicación básica de ejemplo)
- Documentación de la biblioteca

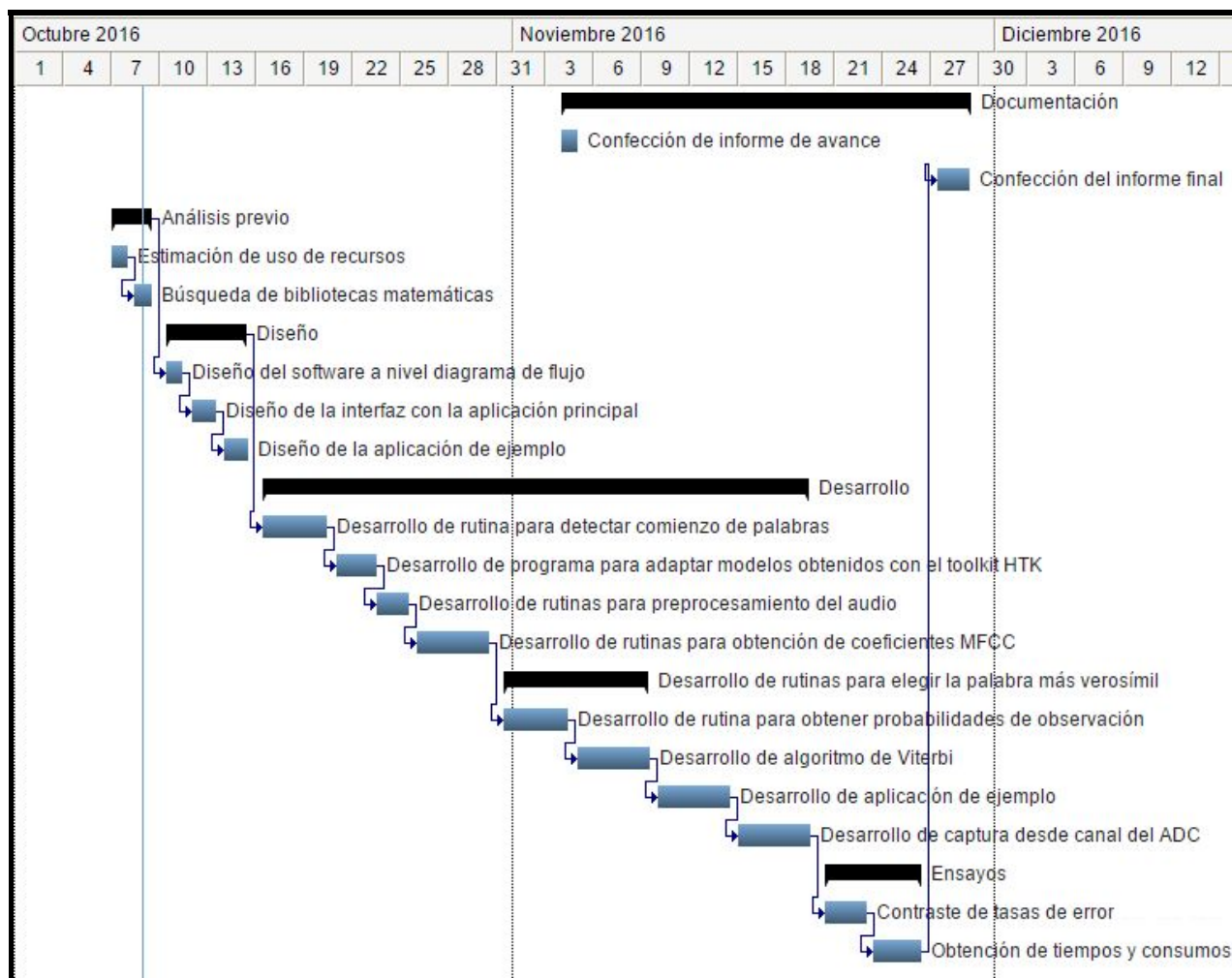
6. Desglose del trabajo en tareas

1. Documentación (9 horas)
 - a. Confección de informe de avance (3 horas)
 - b. Confección del informe final (6 horas)
2. Análisis previo (6 horas)
 - a. Estimación de consumo de memoria y cantidad de procesamiento necesario (3 horas)
 - b. Búsqueda de bibliotecas matemáticas que implementen eficientemente las transformaciones requeridas (3 horas)
3. Diseño (12 horas)
 - a. Diseño del software a nivel diagrama de flujo (4 horas)
 - b. Diseño de la interfaz con la aplicación principal (4 horas)
 - c. Diseño de la aplicación de prueba (4 horas)
4. Desarrollo (70 horas)
 - a. Desarrollo de rutina para detectar comienzo de palabras (10 horas)
 - b. Desarrollo de programa para adaptar modelos obtenidos con el toolkit *HTK* (5 horas)
 - c. Desarrollo de rutinas para preprocesamiento del audio (5 horas)
 - d. Desarrollo de rutinas para obtención de coeficientes *MFCC* (10 horas)
 - e. Desarrollo de rutinas para elegir la palabra más verosímil (20 horas)
 - i. Desarrollo de rutina para obtener probabilidades de observación (10 horas)
 - ii. Desarrollo de algoritmo de *Viterbi* (10 horas)
 - f. Desarrollo de ejemplo de aplicación (10 horas)
 - g. Desarrollo de captura desde canal del *ADC* (10 horas)
5. Ensayos (14 horas)
 - a. Contraste de tasas de error con las obtenidas usando el toolkit *HTK* (7 horas)
 - b. Obtención de tiempos y consumos de memoria de cada una de las rutinas (7 horas)

La suma de todos los tiempos de las tareas resulta en 111 horas.

7. Diagrama de Gantt

	EDT	Nombre	Duración	Inicio	Fin	Predecesoras
1	1	☐ Documentación	13d	03/11/2016	28/11/2016	
2	1.1	Confección de informe de avance	3h	03/11/2016	04/11/2016	
3	1.2	Confección del informe final	6h	26/11/2016	28/11/2016	23
4	2	☐ Análisis previo	1.5d	06/10/2016	08/10/2016	
5	2.1	Estimación de uso de recursos	3h	06/10/2016	07/10/2016	
6	2.2	Búsqueda de bibliotecas matemáticas	3h	07/10/2016	08/10/2016	5
7	3	☐ Diseño	3d	09/10/2016	14/10/2016	
8	3.1	Diseño del software a nivel diagrama de flujo	4h	09/10/2016	10/10/2016	4
9	3.2	Diseño de la interfaz con la aplicación principal	4h	11/10/2016	12/10/2016	8
10	3.3	Diseño de la aplicación de ejemplo	4h	13/10/2016	14/10/2016	9
11	4	☐ Desarrollo	17.5d	15/10/2016	18/11/2016	
12	4.1	Desarrollo de rutina para detectar comienzo de palabras	10h	15/10/2016	19/10/2016	7
13	4.2	Desarrollo de programa para adaptar modelos obtenidos con el toolkit HTK	5h	20/10/2016	22/10/2016	12
14	4.3	Desarrollo de rutinas para preprocesamiento del audio	5h	22/10/2016	24/10/2016	13
15	4.4	Desarrollo de rutinas para obtención de coeficientes MFCC	10h	25/10/2016	29/10/2016	14
16	4.5	☐ Desarrollo de rutinas para elegir la palabra más verosímil	5d	30/10/2016	08/11/2016	
17	4.5.1	Desarrollo de rutina para obtener probabilidades de observación	10h	30/10/2016	03/11/2016	15
18	4.5.2	Desarrollo de algoritmo de Viterbi	10h	04/11/2016	08/11/2016	17
19	4.6	Desarrollo de aplicación de ejemplo	10h	09/11/2016	13/11/2016	18
20	4.7	Desarrollo de captura desde canal del ADC	10h	14/11/2016	18/11/2016	19
21	5	☐ Ensayos	3.5d	19/11/2016	25/11/2016	
22	5.1	Contraste de tasas de error con las obtenidas usando el toolkit HTK	7h	19/11/2016	22/11/2016	20
23	5.2	Obtención de tiempos y consumos de memoria de cada una de las rutinas	7h	22/11/2016	25/11/2016	22



Para lograr la conclusión del proyecto antes de la fecha límite, se estiman que serán requeridas al menos 2 horas diarias de trabajo.

8. Gestión de riesgos

a) Identificación de los riesgos y estimación de sus consecuencias:

Riesgo 1: Dificultad para conseguir la placa de desarrollo

- Severidad (S): 6, porque quizás esta dificultad se traduzca en un retraso a la tarea de probar el funcionamiento del sistema completo.
- Probabilidad de ocurrencia (O): 2, pues hay muchas unidades de la placa disponibles en la facultad.

Riesgo 2: Dificultad para cumplir con los tiempos de las tareas

- Severidad (S): 9, debido a que si se retrasa el proyecto se puede poner en compromiso la aprobación de la materia.
- Probabilidad de ocurrencia (O): 4, pues si bien se estima que será posible realizar la implementación dentro del tiempo esperado puede que alguna tarea se extienda por falta de investigación previa o conocimientos.

Riesgo 3: Que el hardware de la EDU-CIAA no pueda ser usado para el proyecto

- Severidad (S): 10, pues el proyecto no podría ser finalizado cumpliendo su propósito.
- Probabilidad de ocurrencia (O): 3, debido a que cálculos preliminares de cantidad de operaciones matemáticas y memoria RAM requerida resultan en valores mucho menores a los disponibles.

Riesgo 4: Implementar incorrectamente el algoritmo de reconocimiento

- Severidad (S): 8, porque una implementación incorrecta podría hacer que el sistema de reconocimiento no funcione de manera óptima, aumentando excesivamente las tasas de error.
- Probabilidad de ocurrencia (O): 8, debido a que la parte principal del algoritmo se implementará desde cero.

b) Tabla de gestión de riesgos: (El RPN se calcula como $RPN = S \times O$)

Riesgo	Severidad	Ocurrencia	RPN	Severidad*	Ocurrencia*	RPN*
1	6	2	12	6	2	12
2	9	4	36	9	2	18
3	10	3	30	10	1	10
4	8	8	64	8	2	16

Criterio adoptado:

- Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 20

Nota:

- Los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el PRN máximo establecido:

Mitigación de Riesgo 2: Dificultad para cumplir con los tiempos de las tareas

Se harán análisis intermedios del estado de las tareas para poder detectar atrasos en tareas antes de que sea tarde, y en tal caso se podrán modificar levemente los requerimientos para llegar al *deadline*.

- Severidad (S): 9 (no se modifica).
- Probabilidad de ocurrencia (O): 2, se supone que al modificar los requerimientos se podrá llegar con tiempo de sobra a implementar el sistema.

Mitigación de Riesgo 3: Que el hardware de la EDU-CIAA no pueda ser usado para el proyecto

Limitaciones que se pueden encontrar y sus mitigaciones posibles:

- Conversor analógico digital o micrófono demasiado ruidoso → Se podría usar una placa externa con otro conversor, que se comuniquen con el microcontrolador principal mediante una interfaz digital.
- Falta de memoria de datos (RAM) → Se podría intentar usar alguna memoria externa, siempre y cuando la cantidad de pines disponibles sea la suficiente. A cambio seguramente se use más tiempo para leer y escribir datos en la misma.
- Falta de memoria de programa (Flash) → Se podría reducir la complejidad de los modelos utilizados, a costa de mayores tasas de error en el reconocimiento.
- Velocidad de microcontrolador muy lenta → En el peor de los casos, podría significar que el procesamiento tarde demasiado y que el tiempo entre el fin de la palabra y el fin del reconocimiento haga que el sistema completo parezca "lento". Se podría disminuir la complejidad de los modelos para hacerlo más rápido.
- Severidad (S): 10 (no se modifica).
- Probabilidad de ocurrencia (O): 1, por poder cumplir con los objetivos aún si la placa sola no puede ser utilizada.

Mitigación de Riesgo 4: Implementar incorrectamente el algoritmo de reconocimiento

Se realizarán tests por cada módulo implementado, permitiendo así probar el funcionamiento de cada uno de ellos por separado y poder detectar errores de implementación antes de integrarlos.

- Severidad (S): 8 (no se modifica).
- Probabilidad de ocurrencia (O): 2, por realizar tests más seguidos.

9. Gestión de la calidad

Para cada uno de los requerimientos del proyecto indique:

Requerimientos asociados con **adquisición y almacenamiento**

1.1. Tasa de muestreo: 16 ksps

Verificación: se elegirán los valores correctos de período de muestreo para lograrlo.

Validación: se generará externamente una señal audible de frecuencia conocida y se verificará que coincida con la frecuencia de la señal luego de muestreada.

1.2. Bits de muestreo: 10 bits

Verificación: se elegirán los valores correctos de los registros del *ADC*, revisando la hoja de datos del microcontrolador.

Validación: se generará externamente una señal que realice un barrido desde 0 V hasta la tensión de alimentación, y se observará que se guarden aproximadamente 1024 valores distintos.

1.3. Tipo de almacenamiento: continuo, en *buffer* circular

Verificación: se realizará la implementación correcta del mismo, procurando que todas las etapas posteriores puedan funcionar incluso cuando los datos a procesar crucen el fin del *buffer*.

Validación: se grabarán datos arbitrarios y se observará que los datos de hasta dos segundos atrás permanecen en alguna posición de la memoria.

1.4. Duración de almacenamiento: 2 segundos

Verificación: se dimensionará el *buffer* correctamente para que entre el tiempo deseado.

Validación: se grabarán datos arbitrarios y se observará que efectivamente se guardan los datos de los últimos dos segundos.

Requerimientos asociados con **detección de comienzo y fin de palabras**

2.1 Algoritmo a usar: basado en la energía de una ventana pequeña

Verificación: se implementará basándose en el paper "A robust algorithm for accurate endpointing of speech signals" [M.H. Savoji]

Validación: será hecha introduciendo una señal de energía conocida, y verificando que el mismo la estime correctamente.

2.2. Error menor al 10% con una entrada con relación señal-ruido de 25 dB

Verificación: está dentro de los márgenes esperados, según el paper “Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments” [Shen, Hung, Lee]

Validación: se usarán 20 señales de prueba con distintas palabras, a las que se les irá sumando ruido de distinta potencia, hasta llegar a un error en la detección del comienzo y fin.

Requerimientos asociados con **reconocimiento de palabras aisladas**

3.1. Reconocimiento estadístico, basado en cadenas ocultas de Markov y coeficientes MFCC

Verificación: el desarrollo será hecho teniendo en cuenta estos requerimientos.

Validación: se validarán por separado la extracción de coeficientes y el reconocimiento propiamente dicho, comparándolos con los valores obtenidos con el toolkit HTK.

3.2. Soporte de modelos del tipo “mezcla de Gaussianas” con matriz de covarianza diagonal y formato propietario del HTK

Verificación: para lograrlo se desarrollará una aplicación extra que permita convertir dicho formato en uno que sea más fácil de incluir en la memoria de código del microcontrolador. Por ejemplo, podría crearse automáticamente un archivo .c y .h en el que estén todos los datos de los modelos, ya ingresados en las estructuras correspondientes.

Validación: se usarán los modelos hallados en la materia “Procesamiento del Habla”, los cuales deberán ser soportados sin problemas.

3.3. Cantidad máxima de palabras a reconocer: al menos 10 palabras distintas

Verificación: será tenido en cuenta a la hora de la implementación.

Validación: se probará el sistema completo con al menos 10 palabras distintas.

3.4. Cantidad máxima de fonemas por palabra: al menos 10 fonemas por palabra

Verificación: será tenido en cuenta a la hora de la implementación.

Validación: se agregará alguna palabra más larga de lo usual, con al menos 10 fonemas, y se probará que el reconocimiento siga funcionando correctamente.

3.5. Tasas de error dentro de un 5% de las obtenidas con el toolkit HTK

Verificación: por realizar cálculos muy similares a los de otros sistemas, debería cumplirse .

Validación: se compararán dichas tasas de error usando 100 pronunciaciones de distintas palabras, que ya fueron grabadas para la materia anteriormente mencionada.

Requerimientos generales asociados con la **biblioteca**

4.1. Posibilidad de integración a una aplicación existente

Verificación: será implementado como una biblioteca, procurando diseñar una interfaz que sea de fácil uso, que no requiera conocimientos técnicos sobre reconocedores de este tipo.

Validación: será validado al crear un ejemplo de aplicación que funcione correctamente.

4.2. Soporte para obtener las N palabras más probables y una cantidad que las relacione

Verificación: será tenido en cuenta a la hora de elegir el algoritmo de reconocimiento.

Validación: será validado en la misma aplicación de ejemplo, transmitiendo dichos valores por ejemplo a una computadora, para una posterior comprobación manual.