

# TRABAJO FINAL

---

## Proyecto: ApRendizaje

Alumno: Lucios Sjödin – DNI 41.579.975

Tutora: Lorena González

---

El proyecto de trabajo final para este curso simula la posibilidad de que una empresa de Desarrollo de Software cree una aplicación de aprendizaje para el sistema educativo moderno, orientado a funcionar en máquinas de bajos recursos y dando utilidad a las viejas máquinas que permanecen en las salas de informática de jardines y escuelas primarias públicas y/o privadas.

---

## DESCRIPCIÓN DEL PROYECTO Y REQUISITOS FUNCIONALES

---

Un jardín dispone de una vieja sala de informática, con computadoras ya obsoletas. Sin embargo, a los alumnos les interesa usar las computadoras. Para aprovechar el equipo con sus limitaciones, se propone desarrollar una aplicación (en principio de consola) que sirva para educar en aspectos básicos a los niños. Desde operaciones matemáticas simples a cómo cruzar la calle, la aplicación contará con una variedad de 10 temáticas.

### Título: ApRendizaje

(Aplicación de enseñanza básica para Jardín/Escuela Primaria)

# ApRendizaje

Debe proporcionar los siguientes requisitos funcionales:

- Permitir a los docentes un manejo claro del sistema
- Permitir a los niños interactuar mediante menú y submenús en diferentes temas con algunas preguntas para observar lo aprendido
- Proporcionar una interfaz de usuario intuitiva, fácil de utilizar para los y las docentes y sobre todo para los alumnos. Además de la aplicación de consola partiría el siguiente proyecto a desarrollar, otorgarle una interfaz gráfica amigable y atractiva visual y sonoramente.

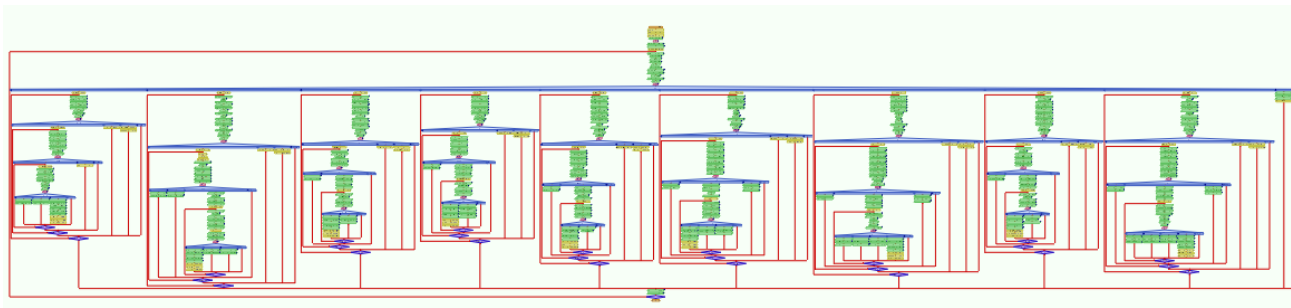
---

## DISEÑO DE INTERFAZ DE USUARIO

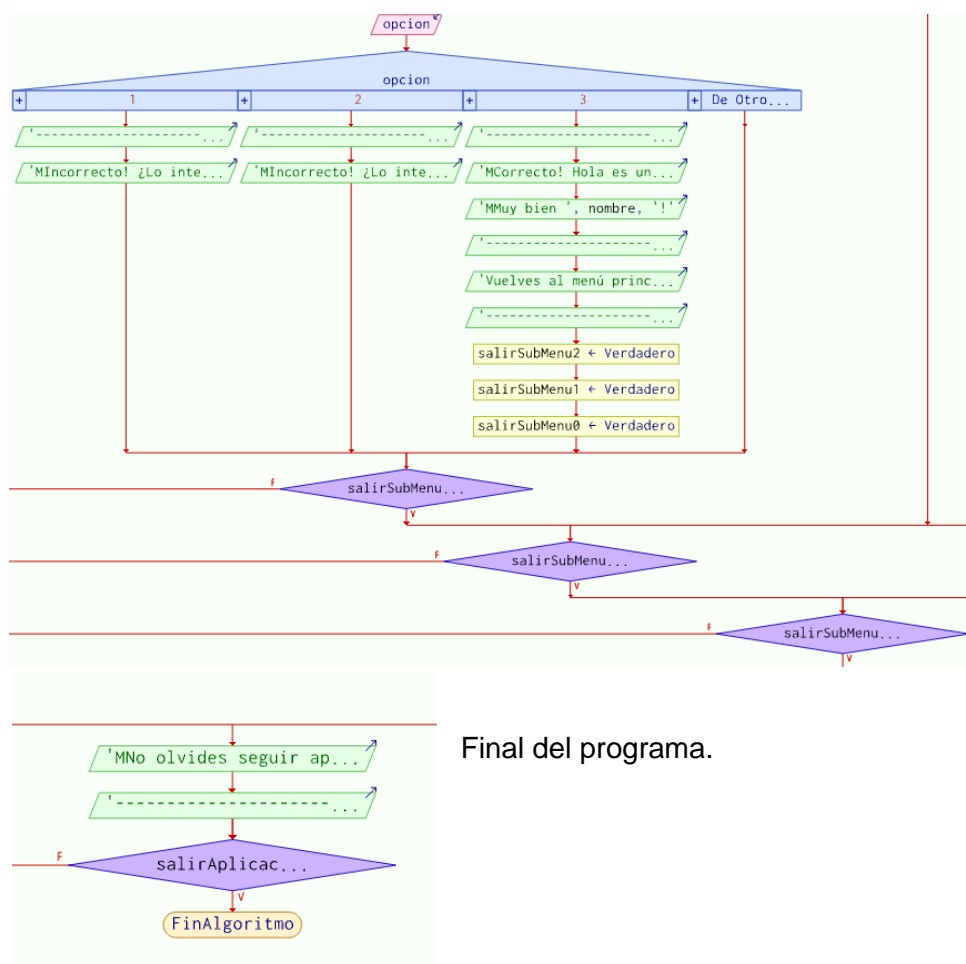
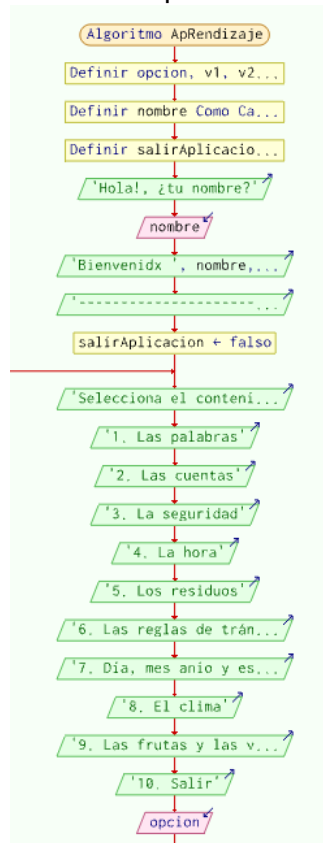
---

Básicamente la interfaz de usuario es una pantalla dónde a cada elección del usuario salen textos y nuevas opciones para una nueva elección. Es una cadena de bucles dentro de otros bucles, ya que, si el usuario elige una opción incorrecta, falla, y el texto anterior se repite y nuevamente debe elegir una opción. Del menú principal de la aplicación se desprenden 10 submenús donde cada uno pertenece a un área del conocimiento distinta y dentro de cada submenú a medida que se avanza se desglosan más submenús para tomas de decisiones del usuario, dando la posibilidad de volver

Aplicación en total:



Ejemplo de final de bucle de un tema elegido:



<https://github.com/RamaLuLuXar/ApiRendizaje>

---

## METODOLOGÍA SCRUM

---

### Product Backlog:

- Registro de usuarios.
- Definición de los recursos educativos (qué contenido se va a enseñar).
- Actividades interactivas.
- Registro de evaluaciones.

### Sprints Planning:

Los Sprints comenzarán siendo de 2 semanas, sin embargo, al final de cada Sprint la revisión sobre la tarea ejecutada podría incidir en una planificación diferente para el próximo Sprint (Sprint Review y Retrospectiva).

Suponiendo que el equipo en su totalidad estuviese solo conformado por una persona, se realizarían Daily Standups (reuniones diarias) para revisar el progreso de las tareas, la posibilidad de mejoras de rendimiento y también los aspectos que complican la ejecución de las tareas.

Entonces, establecer tareas cortas y claras manteniendo un seguimiento regular es fundamental para el desarrollo óptimo del proyecto.

---

## METODOLOGÍA KANBAN

---

Crear en la aplicación llamada **Trello** o alguna de mismas prestaciones, un tablero Kanban que conste de 4 columnas:



(para hacer, haciendo, revisar, hecho)

### Listado de tareas:

- Dividir el proyecto en pequeñas tareas.
- Investigar sobre contenido de enseñanza inicial y primaria e interfaces para niños.
- Diseñar la interfaz de la aplicación.
- Diseñar un diagrama de flujo de la aplicación.
- Escribir un pseudocódigo funcional.

- Definir el contenido de la aplicación.
- Generar una interfaz gráfica y sonora amigable.
- Implementar interacción del usuario.

### **Flujo de trabajo:**

En un principio se colocan todas las tareas del tablero en la columna "PARA HACER". A medida que se comienza a trabajar en cada tarea las mismas son movidas a la columna "HACIENDO". Si el equipo a cargo de la tarea que está haciendo considera que está lista, la tarea se mueve a "REVISAR", dónde se verifica que la tarea esté cumplida de óptima manera. Si la tarea supera la revisión se considera aprobada y terminada, y se ubica en la columna "HECHO".

El tablero Kanban se actualiza constantemente cada día para mantener el progreso de cada tarea bien reflejado en el mismo y que todo el equipo sepa el estado del proyecto.

---

## **COMENTARIOS**

---

El proyecto fue subido a GitHub utilizando la consola local de Git Bash y añadiendo commits desde la consola local en cada actualización del repositorio.

El logo está creado por mí en Photoshop.

Si bien el pseudocódigo es bastante extenso, el mecanismo de cada tema es igual al anterior, bucles Repetir, variables booleanas para entrar y salir de cada lugar y Según opción Hacer.

La idea surgió porque me gustaría que estas cosas básicas aprenda mi sobrina de 5 años, Francesca, en quien me basé para este proyecto.