

Potenciar Argentina

FullCoders: curso introductorio

COMISION 22/21610

Trabajo Final

PROYECTO INTEGRADOR

Título: Plataforma Profesional

Alumna: Bustos, Lorena

DNI: 39687583

Tutor: Amarilla, Luis

Agosto 2023

Objetivo del proyecto

Como el objetivo principal es diseñar y desarrollar una aplicación sencilla utilizando los conceptos vistos en el curso. Propuse la creación de una aplicación para un docente, el cual a través de la carga de datos pueda verificar las condiciones de sus estudiantes, tanto sea por la asistencia como por el promedio al final del cursado.

Por lo que mi necesidad principal es la creación de una aplicación que maneje el porcentaje de asistencias y el promedio luego de tres exámenes.

Por medio de los ejercicios propuestos en los que debíamos calcular porcentajes o verificar condiciones, fue como llegué a la creación del proyecto final.

Desarrollo del proyecto

Un docente requiere verificar las condiciones de sus estudiantes, para eso propuse como requisitos mínimos un porcentaje de asistencia a las clases, y un promedio de los parciales que el profesor decida realizar a lo largo de la cursada, en lo que para aprobar se requiere una calificación promedio mayor o igual a 6 y para promover la materia un puntaje mayor o igual al que el profesor proponga.

El nombre propuesto para la aplicación fue el de: Plataforma Profesional. Con el siguiente logotipo, realizado por medio de namelix.



A su vez, propuse un icono para la visualización de la aplicación de manera sencilla sin necesidad de ver todo el logotipo.



Requisitos Funcionales

Como requisito fundamental, propuse conocer el total de alumnos junto con su nombre para luego poder identificarle a cada uno una asistencia y/o un promedio de calificación. A su vez, el otro requisito inicial también fue conocer el total de clases, para luego facilitar el cálculo del porcentaje, sin necesidad de preguntarlo todas las veces.

Estas peticiones las realicé mediante la aparición de dos pantallas, una pantalla inicial para la carga básica de datos, luego otra para completar los requisitos mínimos de promoción, y finalmente con la aparición de un menú, para facilitar la visualización y que éste nos permita realizar las operaciones las veces necesarias para todos los alumnos que queramos. Los datos los manipulé mediante celdas de acumulación en las que los datos se van registrando para los alumnos a los que le quiera agregar datos. Con una opción final del menú, para poder ver la lista total de todos los alumnos con sus respectivos datos cargados o no.

El sistema proporciona los siguientes requisitos funcionales:

Asistencia:

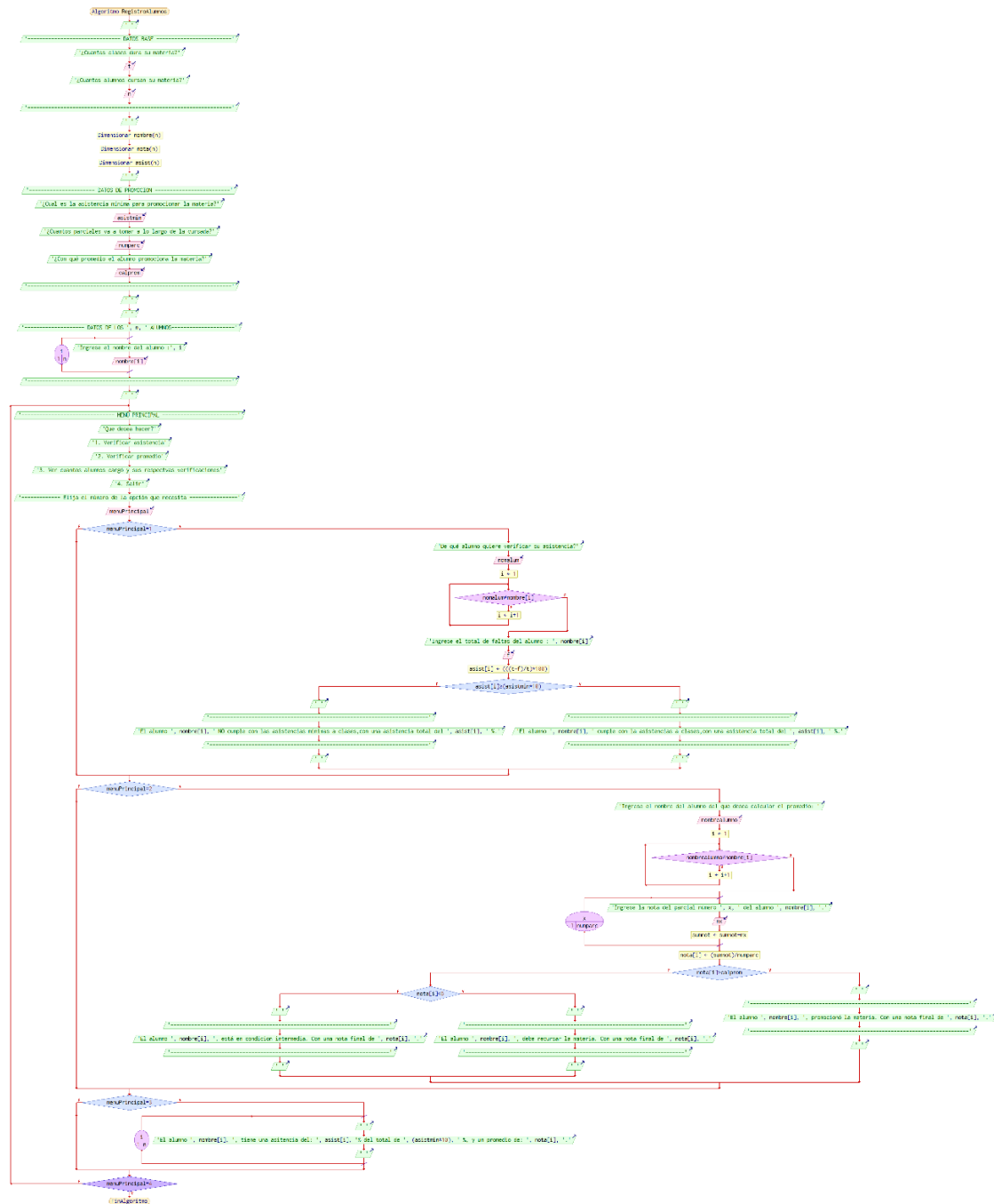
Permite al profesor verificar si el alumno cumple con la asistencia mínima requerida o no.

Promedio de los parciales:

Calcula el promedio de los parciales tomados durante la cursada, logrando que el educador verifique la condición del alumno, en desaprobado, condición intermedio o promovido.

Diseño de Interfaz de Usuario

Mediante la opción del Pseint, cree el diagrama de flujo de la aplicación, para representar la lógica y el flujo de la aplicación.



Desarrollo de Código

El código fue desarrollado por medio de la aplicación recomendada y dada en clases, denominada Pseint. Al final del documento, está el repositorio realizado en GitHub. Pero a continuación está el desarrollo del algoritmo.

Algoritmo PlataformaProfesional

```

Escribir ' '
Escribir '----- DATOS BASE -----'
Escribir '¿Cuántas clases dura su materia?'
Leer t
Escribir '¿Cuántos alumnos cursan su materia?'
Leer n
Escribir '-----'
Escribir ' '

Dimensionar nombre(n)
Dimensionar nota(n)
Dimensionar asist(n)

Escribir ' '
Escribir '----- DATOS DE PROMOCION -----'
Escribir '¿Cuál es la asistencia mínima para promocionar la materia?'
Leer asistmin
Escribir '¿Cuántos parciales va a tomar a lo largo de la cursada?'
Leer numparc
Escribir '¿Con qué promedio el alumno promociona la materia?'
Leer calprom
Escribir '-----'
Escribir ' '

Escribir ' '
Escribir '----- DATOS DE LOS ', n, ' ALUMNOS-----'
```

Para i<-1 Hasta n Hacer

 Escribir 'Ingrese el nombre del alumno :', i

 Leer nombre[i]

FinPara

Escribir '-----'

Escribir ' '

Repetir

 Escribir '----- MENÚ PRINCIPAL -----'

 Escribir 'Que desea hacer?'

 Escribir '1. Verificar asistencia'

 Escribir '2. Verificar promedio'

 Escribir '3. Ver cuantos alumnos cargó y sus respectivas verificaciones'

 Escribir '4. Salir'

 Escribir '----- Elija el número de la opción que necesita -----'

 Leer menuPrincipal

// verificar asistencia

Si menuPrincipal=1 Entonces

 Escribir 'De qué alumno quiere verificar su asistencia?'

 Leer nomalum

 i <- 1

Mientras nomalum<>nombre[i] Hacer

 i <- i+1

 FinMientras

 Escribir 'Ingrese el total de faltas del alumno : ', nombre[i]

 Leer f

 asist[i] <- (((t-f)/t)*100)

Si asist[i]>=(asistmin*10) Entonces

```

        Escribir ' '
        Escribir '-----'
        Escribir 'El alumno ', nombre[i], ' cumple con la asistencias a
clases, con una asistencia total del ', asist[i], ' %.'
        Escribir '-----'
        Escribir ' '
SiNo
        Escribir ' '
        Escribir '-----'
        Escribir 'El alumno ', nombre[i], ' NO cumple con las asistencias
mínimas a clases, con una asistencia total del ', asist[i], ' %.'
        Escribir '-----'
        Escribir ' '
FinSi
FinSi

// verificar promedio
Si menuPrincipal=2 Entonces
    Escribir 'Ingrese el nombre del alumno del que desea calcular el
promedio: '
    Leer nombrealumno

    i <- 1
    Mientras nombrealumno<>nombre[i] Hacer
        i <- i+1
    FinMientras

    Para x<-1 Hasta numparc Hacer
        Escribir 'Ingrese la nota del parcial número ', x, ' del alumno ',
nombre[i], '
        Leer nx
        sumnot <- sumnot+nx

```

FinPara

nota[i] <- (sumnot)/numparc

Si nota[i]>=calprom Entonces

 Escribir ' '

 Escribir '-----'

 Escribir 'El alumno ', nombre[i], ', promocionó la materia. Con una nota final de ', nota[i], ' '

 Escribir '-----'

 Escribir ' '

SiNo

Si nota[i]<6 Entonces

 Escribir ' '

 Escribir '-----'

 Escribir 'El alumno ', nombre[i], ', debe recurrir la materia. Con una nota final de ', nota[i], ' '

 Escribir '-----'

 Escribir ' '

 SiNo

 Escribir ' '

 Escribir '-----'

 Escribir 'El alumno ', nombre[i], ', está en condicion intermedia. Con una nota final de ', nota[i], ' '

 Escribir '-----'

 Escribir ' '

 FinSi

FinSi

FinSi

// Ver los cargados al sistema

Si menuPrincipal=3 Entonces

Para i<-1 Hasta n Hacer

 Escribir ' '

 Escribir 'El alumno ', nombre[i], ', tiene una asitencia del: ',
asist[i], '% del total de ', (asistmin*10), '%, y un promedio de: ', nota[i], '.'

 Escribir ' '

FinPara

FinSi

Hasta Que menuPrincipal=4

FinAlgoritmo

Metodologías Ágiles

La metodología ágil que decidí vincular a mi proyecto es la del Tablero Kanban, en el cual se teoriza sobre el proceso de gestión del proyecto. Además, considero que es la metodología que mejor se adecúa a llevar un relevamiento de datos como lo es ver la condición de los alumnos. En un principio planteé un listado de tareas a realizar para lograr diseñar el pseudocódigo final y luego utilicé Trello para poder plasmarlo.



Listado de Tareas:

- Fijar meta final: Que el docente pueda ver las condiciones finales.
- Identificar los datos a cargar para que el programa comience:
 - Datos básicos:
 - Total de alumnos
 - Total de clases a dictar
 - Datos de promoción
 - Asistencia mínima
 - Cantidad de parciales
 - Promedio mínimo
 - Datos de los alumnos
 - Nombre
- Investigar y diseñar la interfaz del sistema.
- Generar un pseudocódigo funcional. Que dé los siguientes resultados:
 - Condición de asistencia
 - Promedio de los parciales
- Crear la estructura de la base de datos para la carga de la información.
- Implementar la funcionalidad de registro y manejo de los datos generales.
- Implementar la funcionalidad de carga de datos particulares.
- Integrar todas las opciones en una sola opción que muestre todos los datos cargados.

Flujo de Trabajo:

En Kanban cree las columnas "Por hacer", "En progreso", "Revisión" y "Hecho". En un principio colocamos las tareas en la columna "Por hacer". A medida que comenzamos a trabajar en una tarea, la movemos a la columna "En progreso". Cuando completamos una tarea, la movemos a la columna "Revisión" para verificarla y resolver cualquier problema. Una vez que una tarea haya sido revisada y aprobada, la movemos a la columna "Hecho".

Seguimiento:

Actualizaremos el tablero Kanban regularmente, asegurándonos de que refleje el progreso. Cree una columna "Bloqueado" para indicar qué tareas están trabadas y por qué.

Link del tablero creado: <https://trello.com/b/oFW5Z334>

GitHub

En las últimas dos clases, se nos explicó la creación de un repositorio en GitHub para guardar el proyecto. Por lo que toda la documentación se encuentra en el siguiente repositorio GitHub:

<https://github.com/LorenaBustosJ/fullcoders.git>