

Trabajo Final

Potenciar Argentina

FullCoders: Curso

Introductorio

Proyecto: Digit Academy

Repositorio: https://github.com/JuanPabloGiglio/Trabajo_Final.git

Comisión: 21/21661

Nombre: Juan Pablo Giglio

DNI: 42642740

Profesor: Judith Reznik

A large black rectangular box containing the text "DIGIT ACADEMY" in a stylized, colorful font. The word "DIGIT" is in blue, "ACADEMY" is in green, and there is a small pink icon between them.

DIGIT ACADEMY

Descripción del proyecto: "Gestión De Matriculación En Academia"

El proyecto de "Gestión de Matrículas en Academia" es una aplicación diseñada para automatizar y simplificar el proceso de matriculación de estudiantes en una academia educativa. Este sistema tiene como objetivo principal facilitar la inscripción de estudiantes en diferentes cursos, llevar un registro de los datos de los estudiantes y proporcionar estadísticas útiles para la gestión académica.

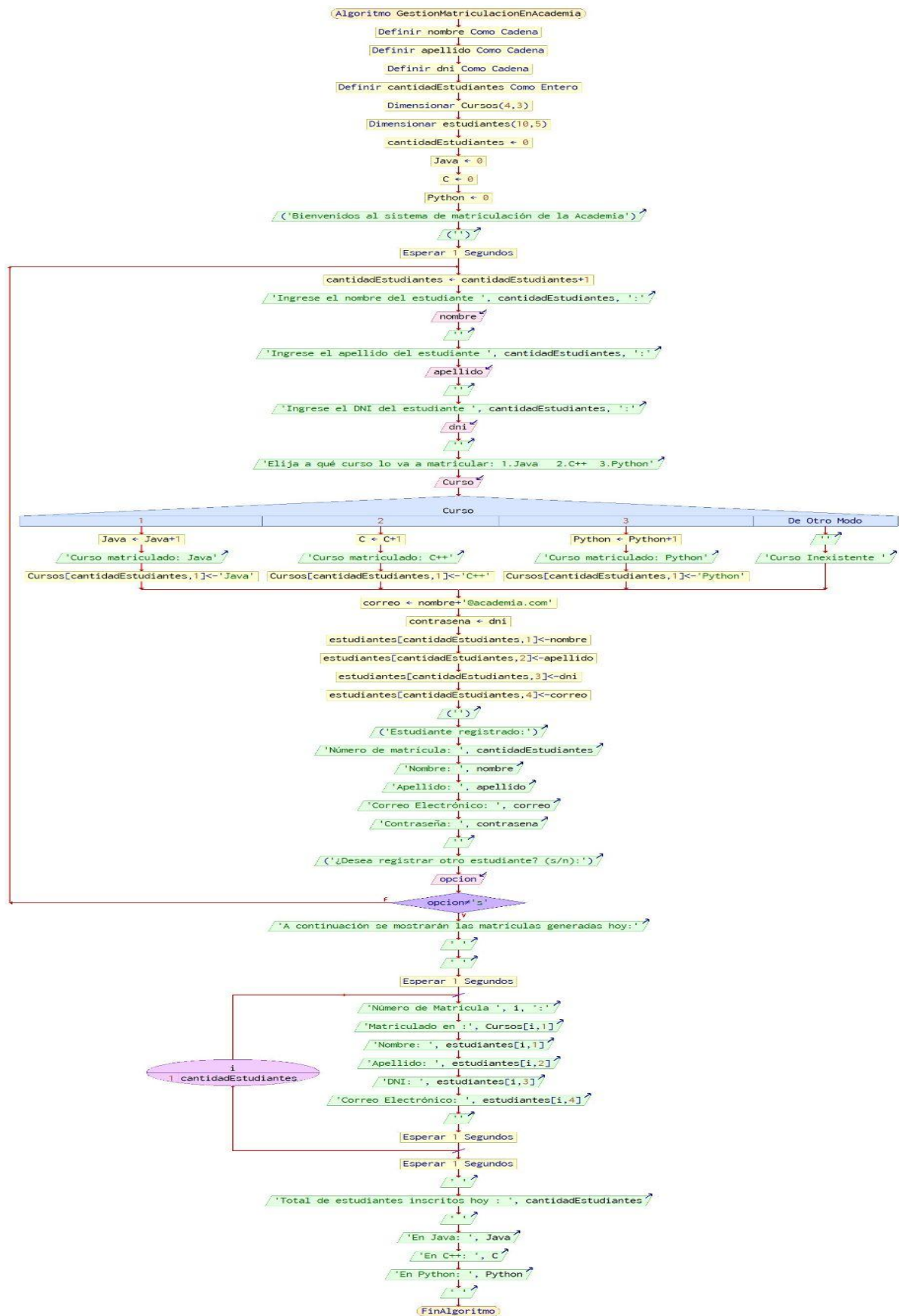
Objetivos del Proyecto:

- 1) Automatización del Proceso de Matriculación: El proyecto busca eliminar la necesidad de realizar procesos manuales y tediosos de inscripción de estudiantes, a través de una interfaz de usuario amigable.
- 2) Registro de Datos de Estudiantes: El sistema almacena de manera segura y organizada información relevante de los estudiantes, incluyendo nombre, apellido, número de DNI, curso al que se matricularon, usuario y contraseña.
- 3) Seguimiento de Matriculaciones por Curso: El proyecto lleva un seguimiento de cuántos estudiantes se han matriculado en cada uno de los cursos ofrecidos por la academia
- 4) Generación de Información de Acceso: Para cada estudiante matriculado, el sistema genera automáticamente una dirección de correo electrónico y una contraseña que les permiten acceder a la academia.
- 5) Informe de Estadísticas: Al finalizar el proceso de matriculación, el sistema presenta un informe detallado que incluye el número total de estudiantes inscritos, así como el desglose de estudiantes matriculados en cada curso específico.

Contenido del Proyecto:

- Diagrama de Flujo General: Representación visual de la lógica del programa.
- Pseudocódigo: Algoritmo del sistema de Matriculación.
- Teoría de Gestión del Proyecto: Exploración de las metodologías ágiles Kanban y Scrum aplicadas al proyecto.

Diagrama de Flujo



Pseudocódigo

```
1  Algoritmo GestionMatriculacionEnAcademia
2  // Declaración de variables
3  Definir nombre Como Cadena
4  Definir apellido Como Cadena
5  Definir dni Como Cadena
6  Definir cantidadEstudiantes Como Entero
7  Dimensionar Cursos[4, 3] // Matriz para almacenar los estudiantes en cada curso
8  Dimensionar estudiantes[10, 5] // Matriz para almacenar datos de estudiantes (nombre, apellido, dni, correo)
9
10 // Inicialización de variables
11 cantidadEstudiantes = 0
12 Java = 0
13 C = 0
14 Python = 0
15
16 // Mensaje de bienvenida
17 Escribir("Bienvenidos al sistema de matriculación de la Academia")
18 Escribir("")
19 Esperar 1 Segundo
20
21 // Bucle principal para registrar estudiantes
22 Repetir
23     cantidadEstudiantes = cantidadEstudiantes + 1
24
25     // Captura de datos del estudiante
26     Escribir "Ingrese el nombre del estudiante ", cantidadEstudiantes, ":"
27     Leer nombre
28     Escribir ""
29
30     Escribir "Ingrese el apellido del estudiante ", cantidadEstudiantes, ":"
31     Leer apellido
32     Escribir ""
33
34     Escribir "Ingrese el DNI del estudiante ", cantidadEstudiantes, ":"
35     Leer dni
36     Escribir ""
37
38     Escribir "Elige a qué curso lo va a matricular: 1.Java 2.C++ 3.Python"
39     Leer Curso
40
41     // Registro del curso matriculado
42     Segun Curso Hacer
43         1:
44             Java = Java + 1
45             Escribir "Curso matriculado: Java"
46             Cursos[cantidadEstudiantes, 1] = "Java"
47         2:
48             C = C + 1
49             Escribir "Curso matriculado: C++"
50             Cursos[cantidadEstudiantes, 1] = "C++"
51         3:
52             Python = Python + 1
53             Escribir "Curso matriculado: Python"
54             Cursos[cantidadEstudiantes, 1] = "Python"
55     De Otro Modo:
56         Escribir ""
57         Escribir "Curso Inexistente "
58     Fin Segun
59
60 // Generación de correo y contraseña
61 correo = nombre + "@academia.com"
62 contraseña = dni
63
64 // Almacenamiento de datos del estudiante
65 estudiantes[cantidadEstudiantes, 1] = nombre
66 estudiantes[cantidadEstudiantes, 2] = apellido
67 estudiantes[cantidadEstudiantes, 3] = dni
68 estudiantes[cantidadEstudiantes, 4] = correo
69
70 // Mensaje de confirmación de registro
71 Escribir("")
72 Escribir("Estudiante registrado:")
73 Escribir "Número de matrícula: ", cantidadEstudiantes
74 Escribir "Nombre: ", nombre
75 Escribir "Apellido: ", apellido
76 Escribir "Correo Electrónico: ", correo
77 Escribir "Contraseña: ", contraseña
78 Escribir ""
79
80 Escribir("¿Debes registrar otro estudiante? (s/n):")
81 Leer opcion
82 Hasta Que opcion = "n"
83
84 // Mostrar lista de estudiantes matriculados por curso
85 Escribir "A continuación se mostrarán las matrículas generadas hoy:"
86 Escribir ""
87 Escribir ""
88 Esperar 1 Segundo
89
90 Para i = 1 Hasta cantidadEstudiantes Hacer
91     Escribir "Número de Matrícula ", i, ":"
92     Escribir "Matriculado en :", Cursos[i, 1]
93     Escribir "Nombre: ", estudiantes[i, 1]
94     Escribir "Apellido: ", estudiantes[i, 2]
95     Escribir "DNI: ", estudiantes[i, 3]
96     Escribir "Correo Electrónico: ", estudiantes[i, 4]
97     Escribir ""
98     Esperar 1 Segundo
99 FinPara
100
101 Esperar 1 Segundo
102
103 // Mostrar resumen de matrículas por curso
104 Escribir ""
105 Escribir "Total de estudiantes inscritos hoy: ", cantidadEstudiantes
106 Escribir ""
107 Escribir "En Java: ", Java
108 Escribir "En C++: ", C
109 Escribir "En Python: ", Python
110 Escribir ""
111
112 FinAlgoritmo
```

Metodología Ágil Scrum

Product Backlog

1. Investigar y diseñar el proyecto: Sistema de Gestión De Matriculación En Academia
2. Diseño de Interfaz de Usuario: Investigar y diseñar la interfaz del sistema amigable.
3. Crear Pseudocódigo: Crear la estructura del Pseudocódigo.
4. Organizar sub-tareas del Pseudocódigo: Asignar las subtareas a las áreas correspondientes
5. Mejorar el pseudocódigo y probarlo: Realizar las mejoras y hacer pruebas exhaustivas.

Sprint Planning (Duración: 2 semanas)

Sprint Goal: "Crear Pseudocódigo"

Daily Standup:

- Ayer: Diseño de interfaz y estructura
- Hoy: Base de Datos
- Obstáculos: Necesitamos saber cómo asignar los cursos.

Sprint Review y Retrospectiva:

- Se logró asignar los cursos.

Retrospectiva:

- Funcionó: asignar los cursos
- Mejorar: Necesito mejorar mi investigación sobre la interfaz para que sea más amigable

Metodología Ágil Kanban

Tablero Kanban

Idea del proyecto

- Sistema de Gestión De Matriculación En Academia
- Investigar y diseñar la interfaz del sistema amigable.

Tareas en progreso

- Organizar sub-tareas del Pseudocódigo
- Crear Pseudocódigo

Tareas en Prueba

- Mejorar el pseudocódigo y probarlo.

Tareas Terminadas

- Gestionar logo de la academia.
- Pseudocódigo fusionable

Seguimiento:

Actualiza tu tablero Kanban regularmente para reflejar el progreso de las tareas. Si alguna tarea se encuentra en estado de demora o enfrenta obstáculos, puedes crear una columna adicional "Demorado" para indicar dónde se está produciendo un retraso y por qué.

