

TRABAJO FINAL

PROYECTO INTEGRADOR

APLICACIÓN PARA GESTIÓN DE ESTACIONAMIENTO DE AUTOS



COMISIÓN 12/21655

SEPTIEMBRE 2023

ALUMNO: LEONARDO SAPEI - DNI: 29.015.478

PROFESOR: PABLO BONADUCE

Objetivo.

Desarrollo de una Aplicación Sencilla: en esta ocasión se eligió una aplicación para gestionar un estacionamiento de automóviles.

Introducción.

En este proyecto, desarrollaremos una aplicación de software en **PSeInt** para gestionar un estacionamiento de automóviles. La aplicación permitirá a los usuarios realizar operaciones como registrar entradas y salidas de vehículos, calcular tarifas de estacionamiento y generar informes.

Aplicaremos conceptos de pensamiento computacional, pseudocódigo, Git/GitHub y metodologías ágiles en nuestra aplicación de gestión de estacionamiento lo que nos ayudará a desarrollar el proyecto de manera más eficiente y efectiva.

Requisitos Funcionales.

Estos requisitos describen las características y funcionalidades básicas de nuestra aplicación. Se describen a continuación:

1. **Registro de Entrada y Salida:** Los usuarios deben poder registrar la entrada y salida de vehículos, capturando la información de la matrícula y la hora.
2. **Cálculo de Tarifas:** La aplicación debe calcular las tarifas de estacionamiento basadas en el tiempo de permanencia del vehículo en el estacionamiento.
3. **Generación de Informes:** Los usuarios deben poder generar informes que muestren la información de los vehículos estacionados, incluyendo matrícula y hora de entrada correspondiente.

Pensamiento Computacional.

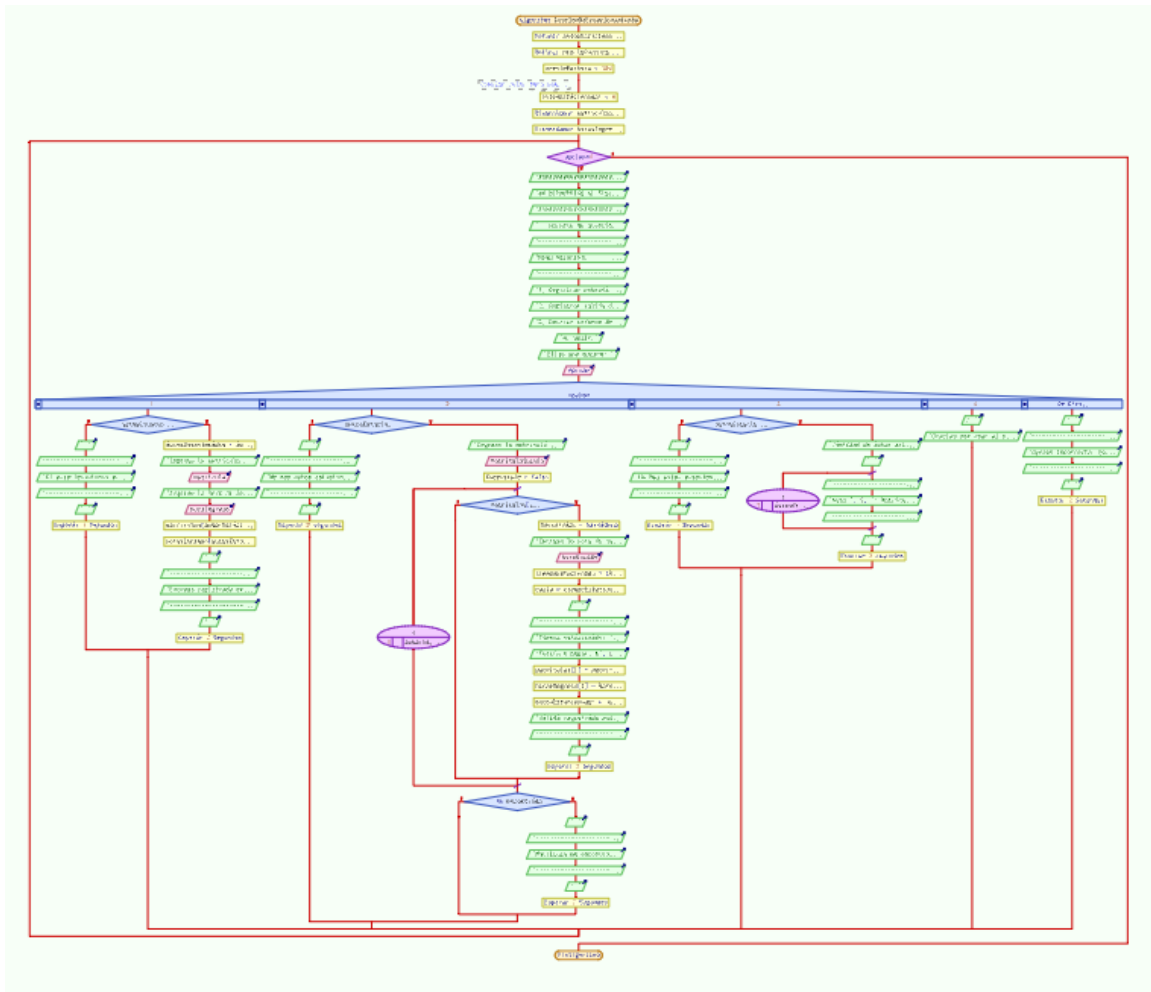
El pensamiento computacional implica dividir un problema en partes más pequeñas y luego desarrollar soluciones para cada parte. En nuestro proyecto de gestión de estacionamiento, aplicamos el pensamiento computacional de la siguiente manera:

- Dividimos el proyecto en partes o módulos: En lugar de abordar todo el proyecto de una vez, identificamos componentes individuales, como el registro de entrada, el registro de salida, la generación de informes, etc. Luego, trabajamos en cada módulo por separado.

- Se definieron algoritmos claros y estructurados: Usamos un enfoque estructurado y algoritmos claros en el código para abordar cada funcionalidad del proyecto. Dividimos las tareas complejas en pasos más pequeños.

- El pseudocódigo es una forma de planificar y diseñar algoritmos antes de escribir el código real. Antes de escribir el código real, se pensó pseudocódigo para cada parte del proyecto. Describiendo la lógica y los pasos que debe seguir el programa.

Diagrama de flujo general.



Diseño de Interfaz de Usuario.

```

#####
## BIENVENIDO AL SISTEMA DE GESTIÓN DE ESTACIONAMIENTO ##
#####

MENÚ PRINCIPAL                                AUTO HOUSE
-----

1. Registrar entrada de vehículo.
2. Registrar salida de vehículo.
3. Generar informe de Vehículos estacionados.
4. Salir.
  
```

Desarrollo de Código.

```

1  Algoritmo GestionDeEstacionamiento
2      Definir autosEstacionados como Real
3      Definir precioPorHora como Real
4      precioPorHora ← 500 // Cambiar valor hora según corresponda
5
6      autosEstacionados ← 0
7
8      Dimension matriculas[100]
9      Dimension horasIngreso[100]
10
11     Mientras opcion ≠ 4
12         Escribir "#####"
13         Escribir "## BIENVENIDO AL SISTEMA DE GESTIÓN DE ESTACIONAMIENTO ##"
14         Escribir "#####"
15         Escribir "    Horario de atención: las 24hs los 365 días del año.  "
16         Escribir "-----"
17         Escribir "MENÚ PRINCIPAL                                AUTO HOUSE"
18         Escribir "-----"
19         Escribir "1. Registrar entrada de vehículo."
20         Escribir "2. Registrar salida de vehículo."
21         Escribir "3. Generar informe de Vehículos estacionados."
22         Escribir "4. Salir."
23         Escribir "Elija una opción: "
24         Leer opcion
25
26     Segun opcion Hacer
27         Caso 1:
28             Si autosEstacionados < 100 Entonces
29                 autosEstacionados ← autosEstacionados + 1
30                 Escribir "Ingrese la matrícula del vehículo: "
31                 Leer matricula
32                 Escribir "Ingrese la hora de ingreso (en formato hhmm): "
33                 Leer horaIngreso
34                 matriculas[autosEstacionados] ← matricula
35                 horasIngreso[autosEstacionados] ← horaIngreso
36                 Escribir " "
37                 Escribir "-----"
38                 Escribir "Entrada registrada exitosamente."
39                 Escribir "-----"
40                 Escribir " "
41                 Esperar 2 segundos
42             Sino
43                 Escribir " "
44                 Escribir "-----"
45                 Escribir "El estacionamiento está lleno."
46                 Escribir "-----"
47                 Escribir " "
48                 Esperar 2 segundos
49             FinSi
50
51         Caso 2:
52             Si autosEstacionados > 0 Entonces
53                 Escribir "Ingrese la matrícula del vehículo que sale: "
54                 Leer matriculaSalida
55                 Encontrado ← Falso
56                 Para i ← 1 Hasta autosEstacionados Hacer
57                     Si matriculas[i] = matriculaSalida Entonces
58                         Encontrado ← Verdadero
59                         Escribir "Ingrese la hora de salida (en formato hhmm): "
60                         Leer horaSalida
61                         tiempoEstacionado ← (horaSalida - horasIngreso[i]) / 100
62                         costo ← tiempoEstacionado * precioPorHora
63                         Escribir " "
64                         Escribir "-----"
65                         Escribir "Tiempo estacionado: ", tiempoEstacionado, " horas"
66                         Escribir "Tarifa a pagar: $", costo
67                         matriculas[i] ← matriculas[autosEstacionados]
68                         horasIngreso[i] ← horasIngreso[autosEstacionados]
69                         autosEstacionados ← autosEstacionados - 1
70                         Escribir "Salida registrada exitosamente."

```

```

71         Escribir "-----"
72         Escribir " "
73         Esperar 2 segundos
74     FinSi
75     FinPara
76     Si No Encontrado Entonces
77         Escribir " "
78         Escribir "-----"
79         Escribir "Matricula no encontrada."
80         Escribir "-----"
81         Escribir " "
82         Esperar 2 segundos
83     FinSi
84     Sino
85         Escribir " "
86         Escribir "-----"
87         Escribir "No hay autos estacionados."
88         Escribir "-----"
89         Escribir " "
90         Esperar 2 segundos
91     FinSi
92
93     Caso 3:
94     Si autosEstacionados > 0 Entonces
95         Escribir "Cantidad de autos estacionados: ", autosEstacionados
96         Escribir " "
97         Para i ← 1 Hasta autosEstacionados Hacer
98             Escribir "-----"
99             Escribir "Auto ", i, ": Matricula ", matriculas[i], " - Hora de ingreso ", horasIngreso[i]
100            Escribir "-----"
101        FinPara
102        Escribir " "
103        Esperar 2 segundos
104    Sino
105        Escribir " "
106        Escribir "-----"
107        Escribir "No hay autos estacionados."
108        Escribir "-----"
109        Escribir " "
110        Esperar 2 segundos
111    FinSi
112
113    Caso 4:
114        Escribir " "
115        Escribir "Gracias por usar el sistema de gestión de estacionamiento."
116
117    De otro Modo:
118        Escribir " "
119        Escribir "-----"
120        Escribir "Opción incorrecta. Por favor, elija una opción válida."
121        Escribir "-----"
122        Escribir " "
123        Esperar 2 segundos
124 +     FinSegun
125     FinMientras
126     FinAlgoritmo

```

GitHub.

Git y GitHub son herramientas esenciales para el control de versiones y la colaboración en proyectos de desarrollo de software. Se creó un repositorio público en GitHub para el proyecto, de manera de poder gestionar el seguimiento de las versiones a medida que se desarrolló el programa.

El link del repositorio es:

<https://github.com/leonardosapei/TRABAJO-FINAL.git>

Metodologías Ágiles.

Se aplicaron metodologías ágiles para gestionar el proyecto de manera individual, dividir el trabajo en tareas más pequeñas, establecer plazos y hacer un seguimiento del progreso de forma fraccionada.

Metodología SCRUM.**Formación del Equipo Scrum.**

Se identifican a todas las personas clave involucradas en el proyecto, como desarrolladores, administradores, usuarios finales u otros interesados.

Creamos el Product Backlog.

Es nuestra lista priorizada de todas las funcionalidades, características, tareas y mejoras que se desean implementar en el proyecto de desarrollo de software. Para el proyecto de gestión de estacionamiento escribimos una lista de elementos que forman parte de nuestro Product Backlog:

1. Registro de entrada de vehículos:

- Permitir a los usuarios registrar la entrada de vehículos.
- Capturar la matrícula del vehículo.
- Capturar la hora de ingreso.
- Validar si hay espacio disponible en el estacionamiento antes de permitir la entrada.

2. Registro de salida de vehículos:

- Permitir a los usuarios registrar la salida de vehículos.
- Capturar la matrícula del vehículo que sale.
- Capturar la hora de salida.
- Calcular el costo del estacionamiento.
- Liberar el espacio ocupado por el vehículo.

3. Generación de informe de vehículos estacionados:

- Mostrar la lista de vehículos estacionados con matrícula y hora de ingreso.

4. Mejoras en la interfaz de usuario:

- Diseñar una interfaz de usuario más amigable y atractiva.

La clave es priorizar estos elementos según su importancia y valor para el cliente y, a medida que se avanza en el desarrollo, agregar, modificar o eliminar elementos del backlog según las necesidades cambiantes del proyecto.

Planificamos el Srpint.

Para organizar los elementos del Product Backlog en sprints con una duración de 15 a 20 días, se consideró la prioridad y la dependencia entre las funcionalidades. Se optó por la siguiente división en Sprints:

Sprint 1:

- Diseñar la interfaz de usuario en PSeInt.
- Implementar el registro de entrada de vehículos.

Sprint 2:

- Implementar el registro de salida de vehículos.
- Calcular tarifas de estacionamiento.

Sprint 3:

- Generar informes de vehículos estacionados.
- Realizar pruebas y ajustes finales.

Durante el desarrollo de los sprints se trabaja en las tareas y actividades definidas en la planificación del sprint para completar los elementos seleccionados.

Se llevan a cabo reuniones diarias para mantener al equipo informado sobre el progreso, los desafíos y las oportunidades de colaboración.

Al final del sprint, el equipo realiza una reunión de revisión del sprint para demostrar las funcionalidades completadas a los interesados y obtener su feedback. También se actualiza el Product Backlog en función de la retroalimentación y los cambios en los requisitos.

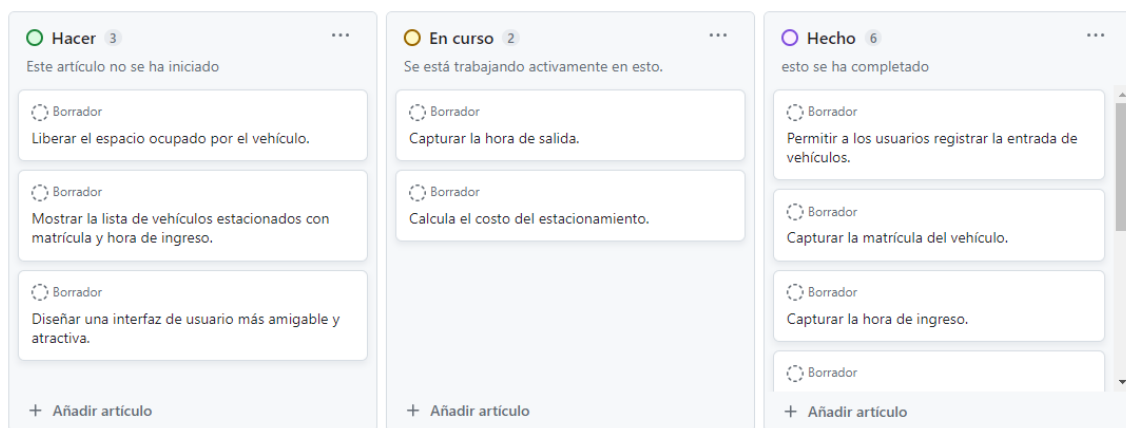
La aplicación de Scrum en el proyecto ayuda a mantener un enfoque iterativo y colaborativo, a responder a las necesidades cambiantes del proyecto y a garantizar una entrega continua de funcionalidades valiosas para los usuarios finales. Además, Scrum promueve la transparencia y la comunicación efectiva entre los miembros del equipo y los interesados.

Metodología KANBAN.

Se planteó el uso de la metodología Kanban como marco de trabajo para la gestión de tareas y el seguimiento del progreso.

Se creó un tablero Kanban, que se podrá visualizar en la solapa "Projects" del repositorio de GitHub. Este tablero se crea para visualizar y priorizar las tareas pendientes y se planteó como objetivo mantener no más de 2 tareas en proceso.

El **flujo de trabajo** para esta metodología es colocar las tareas en la columna "Hacer" del tablero Kanban. Aquí está todo el trabajo que debemos realizar. A medida que comenzamos a trabajar en una tarea, la movemos a la columna "En curso", donde se desarrolla la tarea y revisa para solucionar cualquier problema. Una vez que una tarea haya sido revisada y aprobada, la movemos a la columna "Hecho".



Se deberá realizar un **Seguimiento** y actualización del tablero Kanban regularmente, asegurándonos de que refleje el progreso e identificar qué tareas están demoradas y por qué.

La aplicación efectiva de estos conceptos y herramientas mejora significativamente la gestión, desarrollo y colaboración del proyecto lo que conducirá a un proyecto más exitoso y eficiente.