14-5-2025

Asignatura: Algoritmo

Profesor: Lic. Cáceres Gastón

Nombre Equipo:

Líder:

Integrantes:

Trabajo Práctico N° 2

Trabajo en Equipo

## Introducción

En el presente trabajo práctico, los estudiantes aplicarán los conocimientos adquiridos sobre estructuras básicas de programación utilizando el entorno PSeInt. A través de un problema sencillo y contextualizado, deberán implementar un algoritmo que utilice condicionales, estructuras repetitivas y funciones. Además, se fomentará el trabajo colaborativo mediante el uso de Git y GitHub, permitiendo a los estudiantes experimentar con herramientas profesionales de control de versiones, fortaleciendo tanto sus habilidades técnicas como de trabajo en equipo. Esta experiencia integra conceptos fundamentales de la programación estructurada con buenas prácticas de desarrollo de software.

## Criterios de evaluación

|  |  |  |
| --- | --- | --- |
| CRITERIO | DESCRIPCIÓN | PUNTAJE DEL EQUIPO |
| 1. Correcto uso de estructuras básicas | Uso funcional de PARA, SI, funciones y variables. | ….. / 25 |
| 2. Claridad del código y legibilidad | Código limpio, con comentarios básicos y buena nomenclatura. | ….. / 20 |
| 3. Uso de Git colaborativo | Evidencia de ramas, commits, merge y buen uso de GitHub. | ….. / 25 |
| 4. Cumplimiento de formato y entrega | Archivo con nombre correcto, README.md completo, enlace funcional. | ….. / 15 |
| 5. Participación equitativa | Todos los integrantes deben aparecer en los commits y ramas. | ….. / 15 |
| TOTAL | | ….. / 100 |

## Instrucciones para guardar y entregar el archivo .psc:

El archivo debe llamarse siguiendo esta estructura en **snake\_case**:

**[fecha entrega: aaaammdd]\_[nombre equipo] \_[nombre del proyecto o tema en siglas].extensión**

|  |
| --- |
| Repositorio del Equipo |
|  |

# Control de Ingreso a un Club Recreativo

## Contexto:

Un club recreativo necesita un programa simple para permitir el ingreso de personas según su edad. El sistema deberá preguntar **las veces** que el usuario requiera( por ejemplo 10 personas diferentes) lo siguiente:

* Nombre y edad de la persona.
* Si la persona es menor de 18 años, mostrar: **“Acceso denegado. Solo mayores de edad pueden ingresar.”**
* Si tiene 18 años o más, mostrar: **“Acceso permitido. ¡Bienvenido/a!”**

Al final, debe mostrar:

* Cuántas personas **ingresaron** al club.
* Cuántas fueron **rechazadas** por ser menores de edad.

## Requisitos técnicos obligatorios:

### En el algoritmo (PSeInt), se debe usar:

1. Variables adecuadas para contar accesos permitidos y denegados.
2. Estructura de control PARA para repetir las veces necesarias la carga.
3. Condicional SI...SINO para decidir el acceso.
4. Uso de una **función** que reciba la edad y devuelva Verdadero si puede ingresar, Falso si no.

## Organización y entrega del trabajo

### Requisitos con Git y GitHub:

1. Crear un **repositorio público en GitHub** con el nombre:  
   ClubIngreso2025\_[NombreDelEquipo]

Ejemplo: ClubIngreso2025\_EquipoAlfa

1. Cada integrante debe clonar el repositorio con git clone.

### Crear **una rama por integrante o por módulo de desarrollo**, por ejemplo:

* validacionEdad
* contadorAccesos
* mensajeFinal

0

* dev\_nom\_estudiante1
* dev\_nom\_estudiante2

### Subir los cambios con:

git push origin nombre-de-la-rama y usando los comandos necesarios para el mismo.

Hacer el merge final con la rama **master** antes de la entrega. Cada integrante debe tener configurada el nombre y el correo.