

Universidad de San Carlos de Guatemala

Escuela de Ciencias y Sistemas

Facultad de Ingeniería

Introducción a la programación y Computación 1

Segundo Semestre 2024

Catedrático:

Tutor académico:



FIUSAC
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

PROYECTO 1

Sistema de Gestión de Laboratorio químico para recepción de Muestras

Objetivos

Generales

- Familiarizar al estudiante con el lenguaje de programación Java.
- El estudiante aplique los conocimientos adquiridos en el curso de Introducción a la programación y computación 1.
- Elaborar la lógica para presentar una solución a la problemática planteada.

Específicos

- Utilizar el lenguaje de programación Java como herramienta de desarrollo de software.
- Aplicación de conceptos de programación orientada a objetos.
- Construcción de aplicaciones con interfaz gráfica.
- Implementación de sentencias de control, ciclos, arreglos, matrices y librerías de interfaz gráfica.
- Aplicación de conceptos de programación para crear herramientas administrativas.
- Implementación de soluciones por medio de algoritmos de encriptación.
- Crear respaldos de información y restablecerla con el uso de serialización.

Descripción General

Los laboratorios químicos de IPC Quimik enfrentan múltiples desafíos en la gestión eficiente de muestras y experimentos. Con la recepción constante de muestras de diversas fuentes, es imperativo contar con un sistema robusto que permita registrar cada muestra de manera precisa, asegurando su trazabilidad y estado a lo largo de su ciclo de vida. La identificación única de las muestras, junto con una descripción detallada y el registro de la fecha de recepción, son elementos clave para evitar confusiones y errores que podrían comprometer la integridad de los análisis. Además, la capacidad de modificar y eliminar registros de muestras de manera segura es crucial para mantener una base de datos limpia y actualizada, permitiendo a los investigadores centrarse en el análisis y experimentación sin preocuparse por la integridad de los datos administrativos.

Además de la gestión básica de muestras, existe una necesidad creciente de analizar grandes volúmenes de datos para identificar patrones específicos que son esenciales para la investigación científica y el diagnóstico. La capacidad de detectar la presencia de bacterias o microorganismos en las muestras mediante algoritmos de análisis de datos puede acelerar significativamente el proceso de investigación y permitir la identificación temprana de problemas potenciales. La implementación de un algoritmo de identificación de patrones que utilice técnicas avanzadas como la comparación de vectores y el análisis de similitud es fundamental para manejar la complejidad y el volumen de datos que los laboratorios modernos deben procesar. Este sistema permitirá la carga masiva de datos de muestras y patrones conocidos, y al comparar las muestras con los patrones almacenados, identificará y retornará el nombre del patrón si se detecta una coincidencia. Esto no solo mejora la precisión de los resultados, sino que también proporciona una herramienta poderosa para la investigación y desarrollo en el campo de la química y la biología, apoyando a los investigadores en su búsqueda de soluciones innovadoras y eficientes.

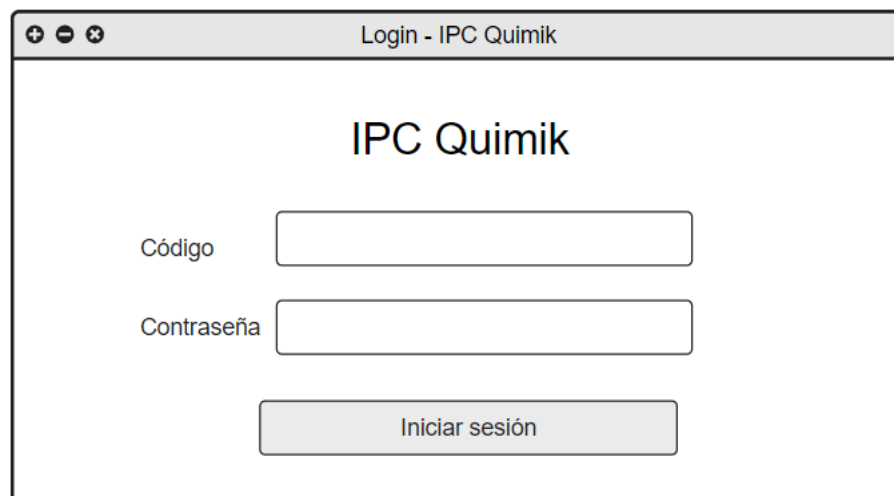
Aplicación

El objetivo principal del sistema es facilitar la gestión eficiente y precisa de las muestras químicas y los experimentos en el laboratorio. Además, se busca proporcionar herramientas avanzadas para el análisis de datos, permitiendo la identificación automática de patrones en grandes volúmenes de datos, lo cual es esencial para la investigación científica y el diagnóstico. Entonces para esto se le solicita a usted como desarrollador, generar las siguientes funcionalidades para ayudar a solucionar el problema del laboratorio de química.

Para desarrollar este sistema es necesario que se desarrolle con el lenguaje de programación Java ya que es un lenguaje de programación robusto, orientado a objetos y muy bueno para aplicar las soluciones necesarias para este problema. Por lo que se requiere que cree las siguientes funcionalidades:

Módulo de Autenticación

La función principal de este módulo es darle el acceso al administrador y a los investigadores por medio de un código y contraseña.

A screenshot of a web browser window titled "Login - IPC Quimik". The window displays the text "IPC Quimik" at the top center. Below this, there are two input fields: the first is labeled "Código" and the second is labeled "Contraseña". Below these fields is a button labeled "Iniciar sesión".

Se tomarán en cuenta las siguientes consideraciones:

- Para poder ingresar como administrador se tendrá un usuario único con el código ***“admin”*** y la contraseña ***“admin”***.
- Este es un sistema cerrado por lo que no hay un registro de usuarios, sino que el administrador debe crear a los investigadores una cuenta para iniciar sesión.
- En caso de que alguno de los datos sea erróneo, se debe mostrar un error.
- Cada vista dentro del sistema debe tener la opción de *cerrar sesión*, la cual regresará a esta primera ventana de login.

Módulo de Administración

Investigadores

En esta pestaña se visualiza la información de los investigadores dentro del sistema. El sistema tiene la capacidad de almacenar varios investigadores.

Código	Nombre	Género	Experimentos
QI-01	Juan Perez	M	5
QI-02	Luis López	M	8

Crear Cargar

Actualizar Eliminar

Top 3 - Investigadores con más experimentos

- Para crear un usuario investigador es necesario contar con un formulario donde se solicite el código (no tiene que ser repetido en el sistema), nombre del investigador, género y contraseña, además los usuarios que son creados en este formulario siempre serán inicializados con 0 experimentos.

Crear Investigador

Código:

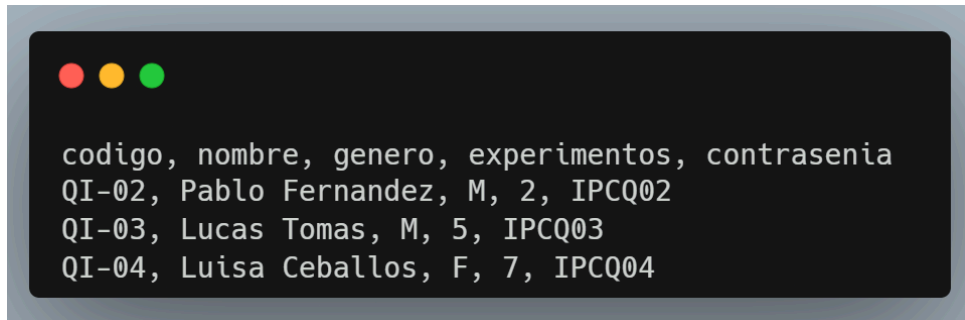
Nombre:

Género:

Contraseña:

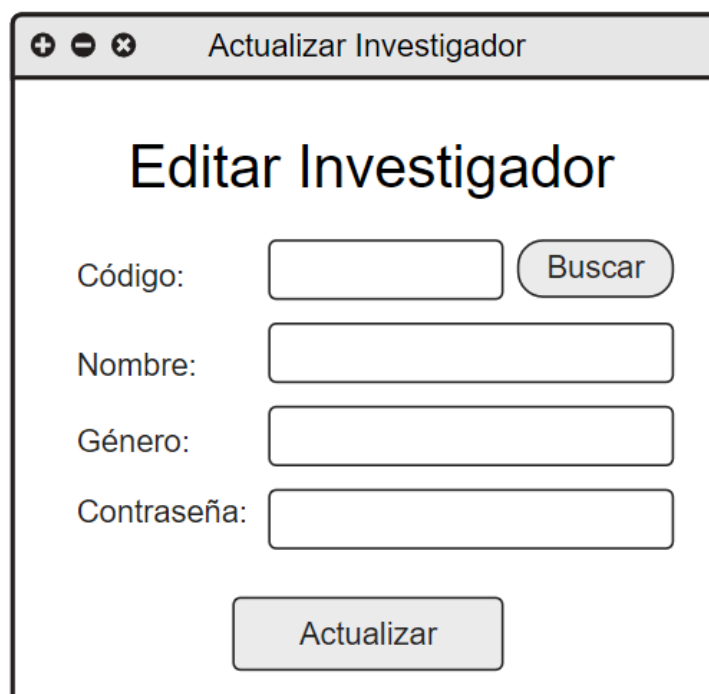
Crear

- Para cargar los usuarios investigadores es necesario cargar un CSV donde contendrá el siguiente formato:



```
codigo, nombre, genero, experimentos, contrasenia
QI-02, Pablo Fernandez, M, 2, IPCQ02
QI-03, Lucas Tomas, M, 5, IPCQ03
QI-04, Luisa Ceballos, F, 7, IPCQ04
```

- Para actualizar los datos del investigador es necesario que use otro formulario donde puede actualizar únicamente el nombre, el género y la contraseña.



Actualizar Investigador

Editar Investigador

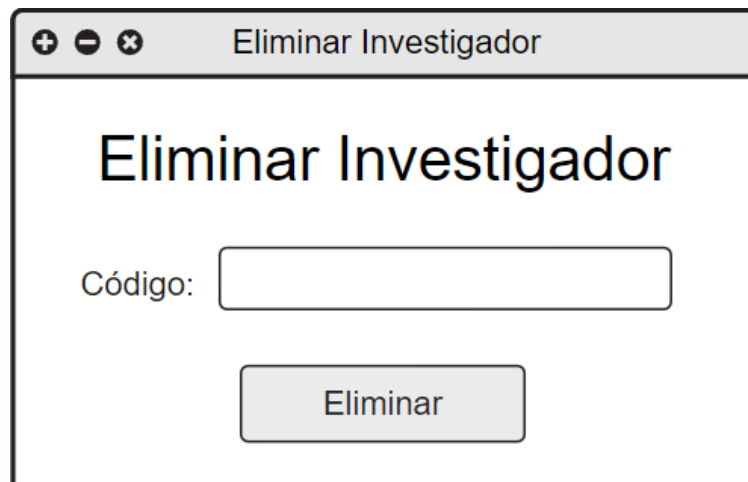
Código:

Nombre:

Género:

Contraseña:

- Para eliminar un investigador únicamente tiene que solicitar el código del investigador para borrarlo del sistema.



Eliminar Investigador

Código:

Muestras

En esta pestaña se visualiza la información de las muestras ingresadas al laboratorio. Este sistema es capaz de almacenar varias muestras para ser investigadas. Inicialmente todas las muestras estarán en estado de *ingreso*.

Código	Descripción	Estado	Acciones
1	Muestra1	Ingreso	<button>Ver</button>
2	Muestra2	En proceso	<button>Ver</button>
3	Muestra3	Procesado	<button>Ver</button>

Crear
Cargar

- Para crear una muestra dentro del sistema, se solicitará el código de la muestra, la descripción de la muestra y se cargará un archivo de tipo CSV del patrón de la muestra. Además iniciará con el estado de "Ingreso".

Crear Muestra

Código:

Descripción:

Patrón:

El patrón a cargar va a ser una matriz de números de tamaño N x N

```
1,4,6,1,5,3
8,6,8,4,1,9
9,3,5,1,9,5
4,3,6,8,2,4
6,3,7,5,8,1
5,4,2,1,5,1
```

- Para cargar las muestras es necesario cargar un CSV donde contendrá el siguiente formato:

```

codigo, descripcion, patron
MQ-02, Muestra 2, 1;4;2;7

```

Estas muestras también tendrán código, descripción y el patrón el cuál estará separado por medio de punto y coma. La raíz cuadrada de la longitud del patrón es el tamaño de filas y columnas de la matriz cuadrada de muestras.

La fórmula de la longitud de filas y columnas de la matriz de muestras en la carga masiva es la siguiente:

$$\text{len}(\text{Filas o Columnas}) = \sqrt{\text{len}(\text{Arreglo patrón separado de punto y coma})}$$

- En el botón de *ver* en la tabla será el encargado de mostrar en HTML la tabla de la matriz de muestras. Este HTML tiene que tener el nombre de `Muestra_<código_muestra>.html`

Asignación de Experimentos

En esta pestaña, el administrador es capaz de asignar las distintas muestras a los distintos investigadores, tiene que verificar si la muestra ya fue asignada para no volverlo a asignar a otro investigador. Además la muestra cambiará su estado a *en proceso*.

The screenshot shows a web application window titled "Administrador - IPC Quimik". It has four tabs: "Investigadores", "Muestras", "Asignación de Experimentos" (which is active), and "Patrones". Inside the active tab, there are two dropdown menus. The first is labeled "Investigador" and has "IQ-01" selected. The second is labeled "Muestra" and has "1" selected. Below these dropdowns is a button labeled "Asignar".

Patrones

En esta pestaña se visualiza la información de los patrones dentro del sistema. El sistema tiene la capacidad de almacenar varios patrones. Estos patrones son las enfermedades y bacterias registradas en el laboratorio.

The screenshot shows a web application window titled "Administrador - IPC Quimik". It has four tabs: "Investigadores", "Muestras", "Asignación de Experimentos", and "Patrones". The "Patrones" tab is active. It contains a table with three columns: "Código", "Nombre", and "Acciones". The table has three rows: the first row contains "PQ-01" and "E. coli." with a "Ver" button; the second row contains "PQ-02" and "Gripe." with a "Ver" button; and the third row is empty. To the right of the table are three buttons: "Crear", "Cargar", and "Eliminar".

Código	Nombre	Acciones
PQ-01	E. coli.	<button>Ver</button>
PQ-02	Gripe.	<button>Ver</button>

Crear Cargar
Eliminar

- Para crear un patrón de bacterias o enfermedades dentro del sistema es necesario solicitar el código, nombre y que cargue el patrón de la enfermedad. El código es único en toda la aplicación.

The screenshot shows a form titled "Crear Patrón". It has three input fields: "Código:", "Nombre:", and "Patrón:". The "Patrón:" field has a "Cargar Patrón" button next to it. At the bottom of the form is a "Crear" button.

Crear Patrón

Código:

Nombre:

Patrón: Cargar Patrón

Crear

El patrón a cargar va a ser una matriz de 1 y 0 de tamaño N x N

```
0,0,1,1,0,1
1,1,1,0,1,0
1,1,0,0,0,1
0,0,0,0,1,1
0,0,1,0,1,1
0,0,1,1,0,1
```

- Para cargar los patrones de bacterias/virus es necesario cargar un CSV donde contendrá el siguiente formato:

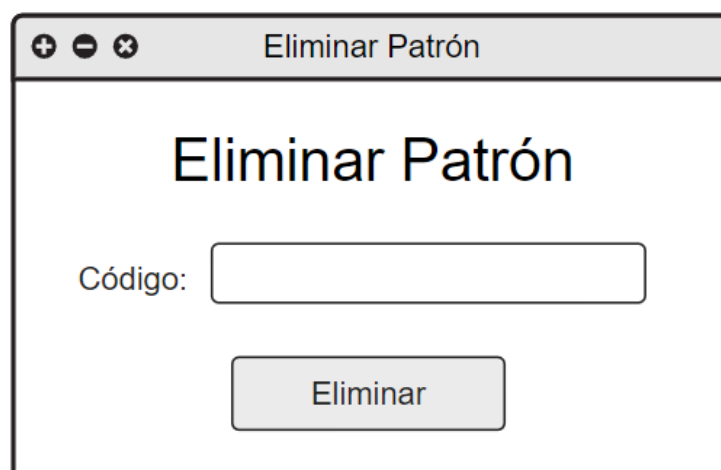
```
codigo, nombre, patron
PQ-03, Gripe, 0;0;1;1
```

Estos patrones también tendrán código, nombre y el patrón el cuál estará separado por medio de punto y coma. La raíz cuadrada de la longitud del patrón es el tamaño de filas y columnas de la matriz cuadrada de muestras.

La fórmula de la longitud de filas y columnas de la matriz de muestras en la carga masiva es la siguiente:

$$\text{len}(\text{Filas o Columnas}) = \sqrt{\text{len}(\text{Arreglo patrón separado de punto y coma})}$$

- Para eliminar un patrón únicamente tiene que solicitar el código del investigador para borrarlo del sistema.



- En el botón de ver en la tabla será el encargado de mostrar en HTML la tabla de la matriz de patrones. Este HTML tiene que tener el nombre de Patrón_<código_patrón>.html

Módulo de Investigadores

Este será el módulo que se mostrará al investigador cuando inicie sesión en el sistema. En este apartado se realizarán los experimentos y se podrán visualizar los resultados de los experimentos hechos por el investigador.

Análisis de Experimentos

En este apartado el investigador realiza los análisis de las muestras haciendo sus experimentos, en la parte de muestra únicamente se mostrarán las muestras asignadas al investigador y puede seleccionar la muestra a analizar. Además puede seleccionar cualquiera de los patrones que se encuentran guardados en el sistema para probar si la muestra coincide con el patrón a analizar mostrará el mensaje de: *La muestra indica que los resultados coinciden con <<nombre_patron>>*, además cambiará el estado de la muestra en *procesado* y ya no se mostrará en el listado de muestras asignadas al investigador. Si la muestra no coincide con el patrón o la longitud del patrón con la muestra no coinciden al analizar, entonces mostrará el siguiente mensaje: *La muestra indica que no coincide con <<nombre_patron>>*. Al terminar de analizar, también abrirá un reporte en HTML del procedimiento realizado en el análisis del experimento y se registrará cada análisis en el sistema y se visualizarán en la pestaña de resultados. El reporte tiene que guardarse con el nombre de AnalisisDDMMMAAAA-HHmms_códigoanálisis.html.

Donde DD es día, MM es mes, AAAA es año, HH es hora, mm es minuto, ss es segundo y códigoanálisis es el código del análisis registrado.

Investigación - IPC Quimik

Investigador IQ-01 Cerrar sesión

Análisis Resultados

Análisis de Experimentos

Muestra

Patrón a analizar

Resultados: La muestra indica Patrón 1

El algoritmo para realizar los análisis de experimentos se detalla más adelante.

Resultados

En esta pestaña se muestran todos los resultados realizados en los análisis de experimentos hechos por el investigador. En la tabla se tendrán que mostrar los siguientes campos: No. Análisis, código de muestra, código de patrón, fecha, hora, resultado, acciones. En la opción de visualizar se tendrá que mostrar el HTML del procedimiento de análisis realizado.

+

-

×

Investigación - IPC Quimik

Investigador IQ-01

Cerrar sesión

Análisis

Resultados

No.	Muestra	Patrón	Fecha	Hora	Resultado	Acciones
23	1	PQ-01	02/08/2024	10:15	Éxito	<div>Ver</div>
55	5	PQ-01	05/08/2024	15:55	Fallo	<div>Ver</div>

ALGORITMO PARA ANÁLISIS DE EXPERIMENTOS

1. Entra la muestra la cual tiene que ser una matriz cuadrada de $N \times N$, la cual está llena de números. En este ejemplo es una matriz de 6×6 la cual llamaremos **MATRIZ 1**.

$$M_1 = \begin{array}{|c|c|c|c|c|c|} \hline 15 & 21 & 9 & 18 & 6 & 6 \\ \hline 18 & 9 & 3 & 27 & 21 & 18 \\ \hline 3 & 9 & 27 & 24 & 18 & 15 \\ \hline 3 & 24 & 27 & 9 & 15 & 12 \\ \hline 3 & 3 & 15 & 24 & 24 & 9 \\ \hline 24 & 15 & 15 & 21 & 21 & 9 \\ \hline \end{array}$$

2. Se multiplica la **MATRIZ 1** con 3 y se guarda en una **MATRIZ TEMP 1**.

$$M_{T1} = M_1 * 3$$

Siguiendo con ese ejemplo el resultado sería:

$$M_{T1} = \begin{array}{|c|c|c|c|c|c|} \hline 45 & 63 & 27 & 54 & 18 & 18 \\ \hline 54 & 27 & 9 & 81 & 63 & 54 \\ \hline 9 & 27 & 81 & 72 & 54 & 45 \\ \hline 9 & 72 & 81 & 27 & 45 & 36 \\ \hline 9 & 9 & 45 & 72 & 72 & 27 \\ \hline 72 & 45 & 45 & 63 & 63 & 27 \\ \hline \end{array}$$

3. Se multiplica la **MATRIZ 2** con 7 y se guarda en una **MATRIZ TEMP 2**.

$$M_{T2} = M_1 * 7$$

Siguiendo con ese ejemplo el resultado sería:

$$M_{T2} = \begin{array}{|c|c|c|c|c|c|} \hline 105 & 147 & 63 & 126 & 42 & 42 \\ \hline 126 & 63 & 21 & 189 & 147 & 126 \\ \hline 21 & 63 & 189 & 168 & 126 & 105 \\ \hline 21 & 168 & 189 & 63 & 105 & 84 \\ \hline 21 & 21 & 105 & 168 & 168 & 63 \\ \hline 168 & 105 & 105 & 147 & 147 & 63 \\ \hline \end{array}$$

4. Se realiza la multiplicación de matrices entre **MATRIZ TEMP 1** y **MATRIZ TEMP 2** y se guardará en una **MATRIZ TEMP 3**.

$$M_{T3} = M_{T1} * M_{T2}$$

Siguiendo con ese ejemplo el resultado sería:

$$M_{T3} = \begin{array}{|c|c|c|c|c|c|} \hline 17766 & 23625 & 23247 & 31185 & 25893 & 19467 \\ \hline 21357 & 30807 & 33264 & 37044 & 34398 & 20790 \\ \hline 16254 & 26082 & 40446 & 40068 & 37800 & 24570 \\ \hline 19278 & 20223 & 30996 & 42903 & 36855 & 25326 \\ \hline 10584 & 21168 & 33264 & 30996 & 30996 & 18522 \\ \hline 21357 & 30996 & 35343 & 43659 & 36477 & 24381 \\ \hline \end{array}$$

5. Se tiene el resultado de la **MATRIZ 2** por medio del resultado de la división modular de 2 con cada uno de los valores de la **MATRIZ TEMP 3**.

$$M_2 = MOD(M_{T3}, 2)$$

Siguiendo con ese ejemplo el resultado sería:

$$M_2 = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

6. Y por último para saber si los resultados son correctos, la **MATRIZ 2** se tiene que comparar con la matriz patrón que se tiene en el sistema y si coincide con la matriz patrón entonces significa que los resultados fueron un éxito, de lo contrario los resultados fueron fallidos.

Serialización

Los datos en el sistema deben persistir aunque la aplicación sea cerrada, por lo que se le solicita utilizar serialización para conservar los datos de una ejecución, y que al iniciar de nuevo la aplicación los datos sigan en el mismo estado de la ejecución anterior.

Se verificará que los archivos en los que se guarde esta información sean archivos binarios, no se permite almacenar estos datos en formato csv, json o cualquier otro tipo. Deben guardarse los objetos serializados en archivos binarios.

Librerías Permitidas

- AWT
- Swing
- LinkedList
- ArrayList
- JFreeChart
- java.io
- java.lang.Math

Requerimientos

Documentación

- Manual Técnico (descripción de los métodos creados y requerimientos de la aplicación, además de diagramas de flujo de los procesos de la aplicación) en PDF.
- Manual de Usuario (Cómo funciona la aplicación y cómo el usuario interactúa con ella) en PDF.
- Diagrama de Clases en PDF.

Restricciones

- La aplicación debe ser desarrollada en el lenguaje de programación JAVA.
- No se permite utilizar código copiado o bajado de internet.
- El IDE por utilizar queda a discreción del estudiante (se recomienda el uso de NetBeans).
- Las copias obtendrán nota de 0 y reporte a la Escuela de Ciencias y Sistemas.
- La interfaz gráfica de usuario puede ser construida con ayuda del IDE (Drag and Drop) o con el uso de las librerías AWT y Swing.
- Durante la calificación se le solicitará al estudiante modificar el código del proyecto con el objetivo de validar la creación de este.
- Su repositorio debe estar privado.
- Cualquier librería que quiera implementar debe consultarlo primero con el auxiliar encargado del curso.
- El estudiante no tendrá derecho a calificación si no presenta interfaz gráfica, no se calificará ninguna funcionalidad en consola.
- El estudiante no tendrá derecho a calificación si no manejó su código dentro de su repositorio privado de github.
- El estudiante no tendrá derecho a calificación si no muestra sus dos hojas de calificación impresas según la fecha y hora escogida.

Habilidades por evaluar

- Uso de variables globales y locales.
- Uso de memoria estática y dinámica.
- Uso de estructuras de control y selección.
- Conocimientos sobre sistemas computacionales.
- Uso de la programación orientada a objetos.
- Habilidad para sintetizar y analizar información.
- Habilidad para comprender y realizar diagramas.
- Habilidad para resolver problemas.
- Capacidad de crear interfaces gráficas de usuario.

Entrega

- **FECHA DE ENTREGA:** 04/09/2024 antes de las 23:59 (No se aceptarán entregas, ni commits a partir de esa fecha y hora).
- En su repositorio adjuntar código fuente y la documentación solicitada.
- El repositorio debe ser privado y en Github, teniendo el nombre de *IPC1_Proyecto1_carnet*, por ejemplo: *IPC1_Proyecto1_202300000*.
- Subir el enlace del repositorio de Github en la tarea asignada en UEDi.
- Agregar de colaborador del repositorio al auxiliar correspondiente a su sección:

Sección	Auxiliar	Usuario de github
A	Rodrigo Alejandro Hernández de León	rodrialeipc
B	Josué Rodolfo Morales Castillo	RMorales202010033
C	David Augusto Maldonado Hurtarte	DavidMaldo02
D	Esteban Humberto Valdez Ennati	Ennox2
E	Douglas Alexander Soch Catalán	DAlexSC
F	Ayeser Cristián Oxlay Juárez	Ayeser-Cristian
G	Federico David Zet Pajoc	fede2510