

## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 01**

Consigna:

1. Registrar un usuario en <https://github.com/>
2. Generar un repositorio en GitHub con el nombre "Computacion - 2018 - [Apellido, Nombre]
3. Generar un repositorio local para todos los trabajos de Computacion de 2018
4. Crear una carpeta Sources/Trabajo Practico 01/Consigna 01
5. Crear una clase Complejo con los atributos
  - a. int real;
  - b. int imag;
6. Escribir los métodos Getter/Setter de todos los atributos
7. Agregar el método input para llenar los atributos del objeto
8. Agregar el método show para mostrar los atributos del objeto
9. Crear una carpeta Sources/Trabajo Practico 01/Consigna 02
10. Crear una clase Persona con los atributos
  - a. long dni;
  - b. string apellido;
  - c. string nombre;
11. Escribir los métodos Getter/Setter de todos los atributos
12. Agregar el método input para llenar los atributos del objeto
13. Agregar el método show para mostrar los atributos del objeto
14. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 02**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 02/Consigna 01
2. Crear una clase Fecha con los atributos
  - a. int dia;
  - b. int mes;
  - c. int anho;
3. Escribir los métodos Getter/Setter de todos los atributos
4. Escribir un constructor que inicialice los atributos a una fecha #01/01/1980#
5. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
6. Agregar el método input para llenar los atributos del objeto
7. Agregar el método show para mostrar los atributos del objeto
8. Agregar un método que permita comparar dos objetos Fecha
9. Agregar un método que permita sumarle días a un objeto Fecha
10. Crear una carpeta Sources/Trabajo Practico 02/Consigna 02
11. Crear una clase Persona con los atributos
  - a. long dni;
  - b. string apellido;
  - c. string nombre;
  - d. Fecha nacimiento;
12. Escribir los métodos Getter/Setter de todos los atributos
13. Escribir un constructor que inicialice los atributos vacíos
14. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
15. Agregar el método input para llenar los atributos del objeto
16. Agregar el método show para mostrar los atributos del objeto
17. Crear una carpeta Sources/Trabajo Practico 02/Consigna 03
18. Crear una clase Comunidad con el atributo
  - a. Persona personas[10];
19. Escribir los métodos Getter/Setter
20. Escribir un constructor que inicialice los atributos vacíos
21. Agregar el método input para llenar los atributos del objeto
22. Agregar el método show para mostrar los atributos del objeto
23. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 03**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 03/Consigna 01
2. Crear una clase Tool con los siguientes métodos:
  - a. `int incrementByValue(int);`
  - b. `void incrementByReference(int*);`
  - c. `int decrementByValue(int);`
  - d. `void decrementByReference(int*);`
  - e. `void orderArrayByReference(int*, int size);`
  - f. `int* orderArrayAndCopy(int*, int size);`
  - g. `void showArray(int*, int size);`
3. Crear una carpeta Sources/Trabajo Practico 03/Consigna 02
4. Crear una clase Matrix con los atributos
  - a. `int columns;`
  - b. `int rows;`
  - c. `int* values;`
5. Escribir los métodos Getter/Setter de todos los atributos
6. Escribir un constructor que inicialice los atributos vacíos
7. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
8. Agregar los métodos
  - a. `void inputByArrayMethod();`
  - b. `void inputByMemoryMethod();`
9. Agregar los métodos
  - a. `void showByArrayMethod();`
  - b. `void showByMemoryMethod();`
10. Agregar los métodos
  - a. `Matrix addByArrayMethod(Matrix);`
  - b. `Matrix addByMemoryMethod(Matrix);`
11. Agregar los métodos
  - a. `Matrix multiplyByArrayMethod(Matrix);`
  - b. `Matrix multiplyByMemoryMethod(Matrix);`
12. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 04**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 04/Consigna 01
2. Crear una clase Fecha con los atributos
  - a. int dia;
  - b. int mes;
  - c. int anho;
3. Escribir los métodos Getter/Setter de todos los atributos
4. Escribir un constructor que inicialice los atributos a una fecha #01/01/1980#
5. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
6. Agregar el método input para llenar los atributos del objeto
7. Agregar el método show para mostrar los atributos del objeto
8. Agregar un método que permita comparar dos objetos Fecha
9. Agregar un método que permita sumarle días a un objeto Fecha
10. Crear una carpeta Sources/Trabajo Practico 04/Consigna 02
11. Crear una clase Persona con los atributos
  - a. long dni;
  - b. string apellido;
  - c. string nombre;
  - d. Fecha\* nacimiento;
12. Escribir los métodos Getter/Setter de todos los atributos
13. Escribir un constructor que inicialice los atributos vacíos
14. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
15. Agregar el método input para llenar los atributos del objeto
16. Agregar el método show para mostrar los atributos del objeto
17. Crear una carpeta Sources/Trabajo Practico 04/Consigna 03
18. Crear una clase Comunidad con el atributo
  - a. Persona\* personas;
  - b. int size\_personas;
19. Escribir los métodos Getter/Setter
20. Escribir un constructor que inicialice los atributos vacíos
21. Agregar el método input para llenar los atributos del objeto
22. Agregar el método show para mostrar los atributos del objeto
23. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 05**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 05/Consigna 01
2. Crear una clase Fecha con los atributos
  - a. int dia;
  - b. int mes;
  - c. int anho;
3. Escribir los métodos Getter/Setter de todos los atributos
4. Escribir un constructor que inicialice los atributos a una fecha #01/01/1980#
5. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
6. Crear una clase Persona con los atributos
  - a. long dni;
  - b. string apellido;
  - c. string nombre;
  - d. Fecha\* nacimiento;
  - e. int legajo;
7. Escribir un programa que permita agregar, modificar, buscar, listar y eliminar datos de la clase Persona en un archivo de textos, se usará el legajo como dato clave de las personas.
8. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 06**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 06/Consigna 01
2. Implementar una pila que permita trabajar con datos de tipo double
3. Crear una carpeta Sources/Trabajo Practico 06/Consigna 02
4. Crear una clase Fecha con los atributos
  - a. int dia;
  - b. int mes;
  - c. int anho;
5. Escribir los métodos Getter/Setter de todos los atributos
6. Escribir un constructor que inicialice los atributos a una fecha #01/01/1980#
7. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
8. Implementar una pila que permita trabajar con punteros a datos de tipo Fecha
9. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

**Guía de Trabajos Prácticos 2018**

**Trabajo Práctico 07 (Parcial 1)**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 07/Consigna 01
2. Generar un archivo de texto que contenga intercalados los datos de Alumno y de Examen con las siguientes características
  - a. Las líneas de Alumno comienzan con #alumno
  - b. Las líneas de Examen comienzan con #examen
  - c. De Alumno guardar nombre, apellido, legajo
  - d. De Examen guardar materia, fecha, nota, libro, folio
3. Los alumnos serán cargados en memoria en una Cola
4. Cada nodo de la cola tendrá un puntero a una Pila que contiene todos los objetos Examen
5. La Cola de Alumno será un atributo de la clase Curso
6. El dato fecha de Examen se guardará como string
7. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

**Guía de Trabajos Prácticos 2018**

**Parcial 1 - Presencial**

Consigna:

1. Crear una carpeta Sources/Parcial 01/Consigna 01
2. Leer un archivo de texto con palabras
3. Cada palabra del archivo alimentará una pila
4. Cada nodo de la pila además apuntará a una cola que contendrá en cada nodo cada letra de la palabra
5. Hacer un método show en la clase pila que muestre todas las palabras de los nodos y por cada nodo que muestre las letras contenidas en la cola
6. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto



## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 08**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 08/Consigna 01
2. Implementar una lista simple que permita trabajar con datos de tipo double
3. Crear una carpeta Sources/Trabajo Practico 08/Consigna 02
4. Crear una clase Fecha con los atributos
  - a. int dia;
  - b. int mes;
  - c. int anho;
5. Escribir los métodos Getter/Setter de todos los atributos
6. Escribir un constructor que inicialice los atributos a una fecha #01/01/1980#
7. Sobrecargar el constructor para que se pueda inicializar el objeto con valores enviados
8. Implementar una lista simple que permita trabajar con punteros a datos de tipo Fecha
9. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto

## **Guía de Trabajos Prácticos 2018**

### **Trabajo Práctico 09 (Parcial 2)**

Consigna:

1. Crear una carpeta Sources/Trabajo Practico 09/Consigna 02
2. Generar un archivo de texto que contenga intercalados los datos de Materia, de Alumno y de Examen con las siguientes características
  - a. Las líneas de Materia comienzan con #materia
  - b. Las líneas de Alumno comienzan con #alumno
  - c. Las líneas de Examen comienzan con #examen
  - d. De Materia guardar nombre, carrera, curso, profesor
  - e. De Alumno guardar nombre, apellido, legajo
  - f. De Examen guardar materia, fecha, nota, libro, folio
3. Las materias serán cargadas en memoria en una Lista
4. Cada nodo de la lista tendrá un puntero a una Cola que contiene todos los objetos Alumno que la cursan
5. Cada nodo de la cola tendrá un puntero a una Pila que contiene todos los objetos Examen
6. La lista de Materia será un atributo de la clase Carrera
7. El dato fecha de Examen se guardará como string
8. Construir en Carrera un método show que muestre todos los datos
9. Hacer un git push de todas las modificaciones del repositorio local en el repositorio remoto