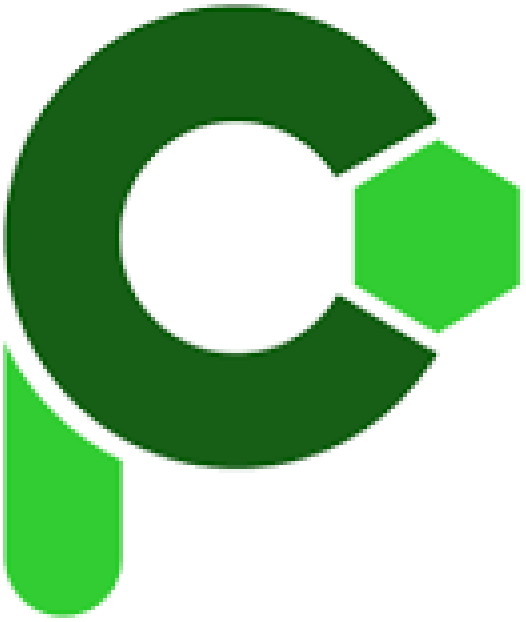
Pràctica entorns



INDEX

1.-Configuración git

Configura GIT para el proyecto. Crea un repositorio público en GitHub y conectarlos.
-git init

```
PS C:\Users\rikip\Documents\JAVA\Entorns\Pràctica> git init
```

-git remote add

```
PS C:\Users\rikip\Documents\JAVA\Entorns\Pràctica> git remote add origin https://github.com/rikipove/practicaEntorns.git
```

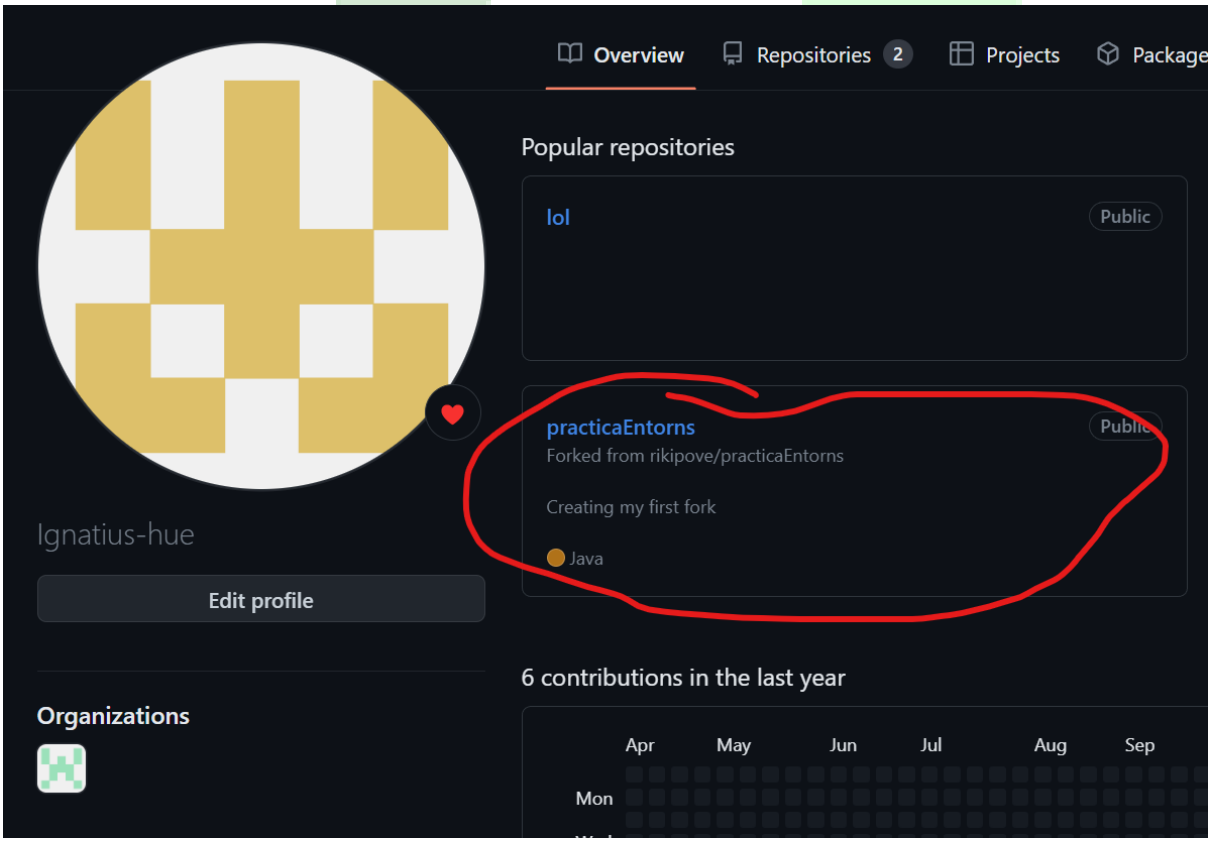
-git push to save changes

```
PS C:\Users\rikip\Documents\JAVA\Entorns\Pràctica> git commit -m "1st push"
[master (root-commit) c95f619] 1st push
 2 files changed, 171 insertions(+)
 create mode 100644 registroUsuario.java
 create mode 100644 validarCampos.java
```

2.-Fork

Realiza un fork del repositorio del compañero de práctica y realiza, al menos, una operación commit, push, pull y pull request, comentando el resultado de cada ejecución. (1 punto)

Fork:



-git clone para poder llevar a cabo las operaciones de commit, push, pull y pull request:

```
C:\Users\ignas\Desktop\fork>git clone https://github.com/rikipove/practicaEntorns.git
Cloning into 'practicaEntorns'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

-git add .

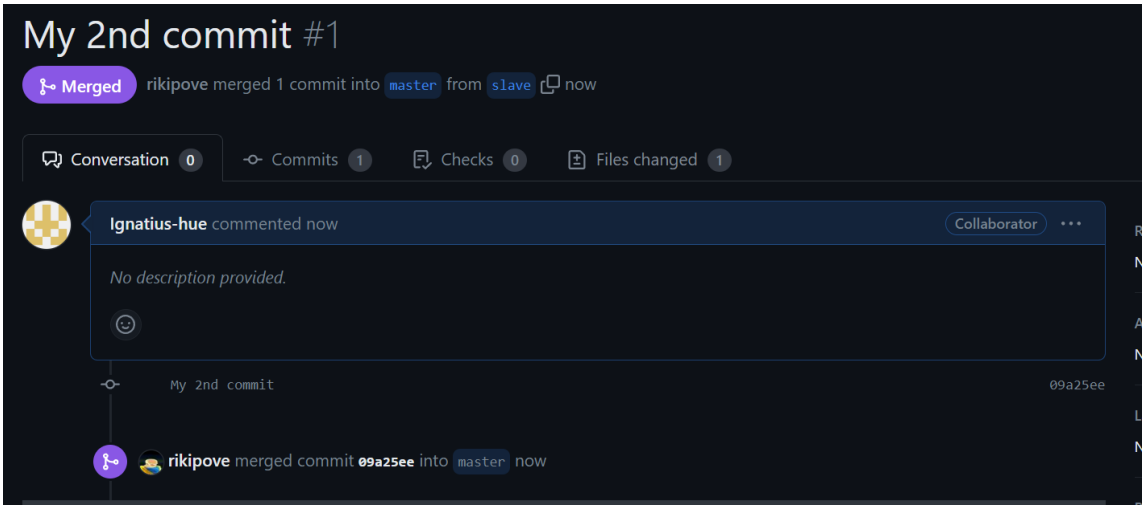
-git commit

```
C:\Users\ignas\Desktop\fork>git add .
warning: adding embedded git repository: practicaEntorns
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> practicaEntorns
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached practicaEntorns
hint:
hint: See "git help submodule" for more information.

C:\Users\ignas\Desktop\fork>git commit -m "My first commit"
[master (root-commit) 6c662f6] My first commit
 1 file changed, 1 insertion(+)
 create mode 160000 practicaEntorns
```

git push

```
PS C:\Users\ignas\Desktop\fork2\practicaEntorns> git push --set-upstream origin slave
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/rikipove/practicaEntorns.git
   ccbb3ab..09a25ee  slave -> slave
branch 'slave' set up to track 'origin/slave'.
```



3.-Historial de versiones

Muestra el historial de versiones para el proyecto mediante un comando desde consola y visualiza los cambios realizados por cada miembro de la práctica. (1 punto)

```
PS C:\Users\rikip\Documents\JAVA\Entorns\Pràctica> git log --oneline
09a25ee (HEAD -> master, origin/slave, slave) My 2nd commit
ccbb3ab (origin/master) refactor
c95f619 1st push
PS C:\Users\rikip\Documents\JAVA\Entorns\Pràctica> git history
git: 'history' is not a git command. See 'git --help'.
PS C:\Users\rikip\Documents\JAVA\Entorns\Pràctica> git log
commit 09a25ee78b47b6614e5c65c1012c79d443e56963 (HEAD -> master, origin/slave, slave)
Author: Miquel Jaume <BMW_lover@gmail.com>
Date:   Wed Mar 29 20:10:47 2023 +0200

    My 2nd commit

commit ccbb3ab35b59ca1bcd2f53da77f730193c3de9bb (origin/master)
Author: Casamonstros <algo@gmail.com>
Date:   Wed Mar 29 19:39:00 2023 +0200

    refactor

commit c95f619c88dcc41fc9f132b9ff6c2dbdaf5407d0
Author: Casamonstros <algo@gmail.com>
Date:   Wed Mar 29 17:41:01 2023 +0200

    1st push
PS C:\Users\rikip\Documents\JAVA\Entorns\Pràctica>
```

4.-Cambio nombre variable

Cambia el nombre de la variable que recoge el campo usuario por "nombreDeUsuario" y ponla como parámetro del método que lo valida. (1 punto)

```
public class registroUsuarioNuevo {
    static String nombreDeUsuario = "";
    static String email = "";
    static String password = "";
    static String code = "";
    static Scanner sc = new Scanner(System.in);
    static validarCamposNuevo vc = new validarCamposNuevo();

    // name input
    public static void nameInput() {
        while (!vc.nameCheck(nombreDeUsuario)) {
            System.out.println(x:"Introduce your name: ");
            nombreDeUsuario = sc.nextLine();
        }
```

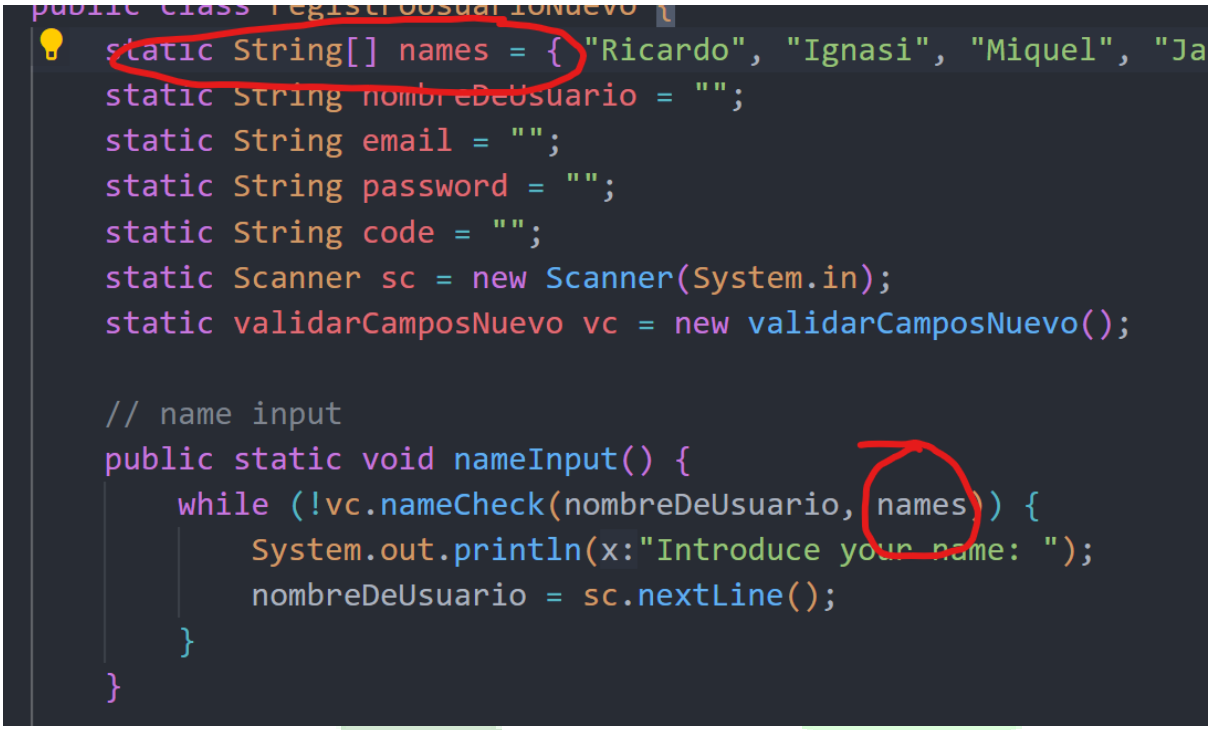
5.-Método init

Introduce el método init en la clase registroUsuario, que englobe las sentencias de la clase registroUsuario que operan con el objeto validaUsuario. (1 punto)

```
    public static void init(){
        //inicialitzem els mètodes
        nameInput();
        emailInput();
        passwordInput();
        codeInput();
    }
    Run | Debug
    public static void main(String[] args) {
        // Inicialització
        init();
    }
}
```

6.-Nuevo parámetro

Añade un nuevo parámetro al método compruebaNombre, de nombre “usuarios” y de tipo array de Strings o cambia el orden de los parámetros de los métodos si tienen más de uno. (1 punto)



```
public class registroUsuarioNuevo {
    static String[] names = { "Ricardo", "Ignasi", "Miquel", "Ja
    static String nombreDeUsuario = "";
    static String email = "";
    static String password = "";
    static String code = "";
    static Scanner sc = new Scanner(System.in);
    static validarCamposNuevo vc = new validarCamposNuevo();

    // name input
    public static void nameInput() {
        while (!vc.nameCheck(nombreDeUsuario, names)) {
            System.out.println(x:"Introduce your name: ");
            nombreDeUsuario = sc.nextLine();
        }
    }
```

7.-Constructores

Crea constructores y encapsula atributos. (1 punto)

```java
public class validarCamposNuevo {

    String[] domains;
    String letras;
    String numeros;
    String simbolos;

    /**
     * Método que comprueba si los campos introducidos son válidos y el código de
     * seguridad generado coincide con el introducido por el usuario.
     *
     * @param name     Nombre del usuario.
     * @param email    Dirección de correo electrónico del usuario.
     * @param password Contraseña elegida por el usuario.
     * @param code     Código de seguridad introducido por el usuario.
     * @return True si todos los campos son válidos y los códigos coinciden, False
     *         en caso contrario.
     */

    public validarCamposNuevo() {
        this.domains = new String[] { "paucasesnovescifp", "yahoo", "gmail", "hotmail" };
        this.letras = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
        this.numeros = "0123456789";
        this.simbolos = "!@#$%^&*()_+-={}[]\\|:;\"'<>,.?/";
    }
```

8.-Comentarios

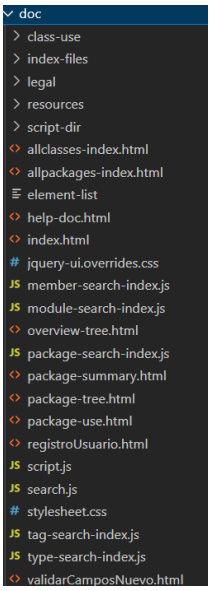
Inserta comentarios en la clase validarCampos y registroUsuario. (1 punto)

```
//encapsulamiento de atributos
public validarCamposNuevo() {
    this.domains = new String[] { "
    this.letras = "abcdefghijklmnop
    this.numeros = "0123456789";
    this.simbolos = "!@#$%^&*()_+-=
}
```

```
//método init donde agregamos los métodos a ejecutar
public static void init() {
    // inicialitzem els mètodes
    nameInput();
    emailInput();
    passwordInput();
    codeInput();
}
```

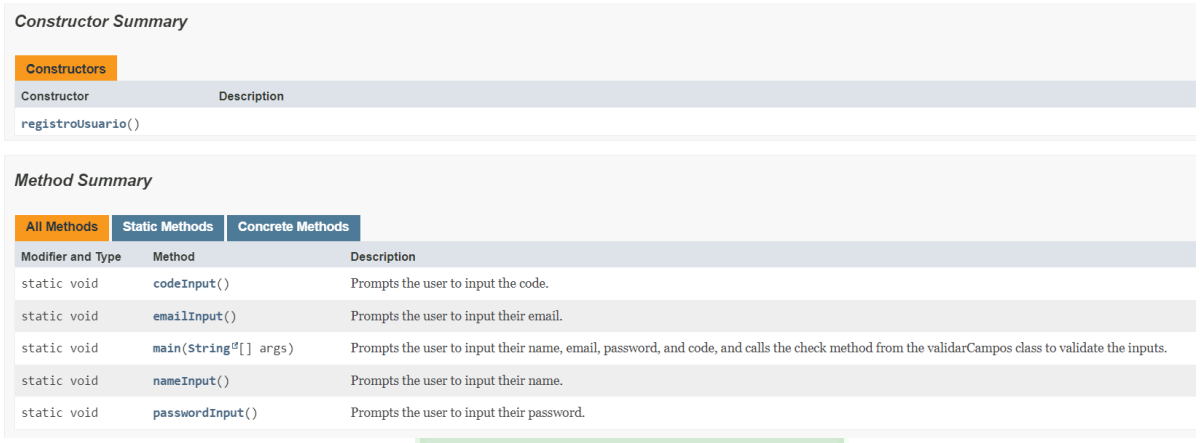
9.-Documentación javadoc

Genera documentación Javadoc para todo el proyecto. (1 punto)



10.-Comprobación

Comprueba que la documentación generada por Javadoc, abarca todos los métodos y/o atributos de la clase validarCampos y el método principal de la clase registroUsuario. (1 punto)



Constructor Summary

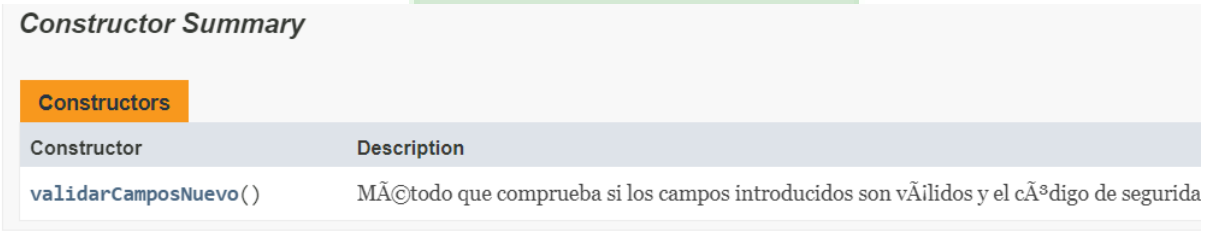
Constructors

Constructor	Description
`registroUsuario()`	

Method Summary

All Methods | Static Methods | Concrete Methods

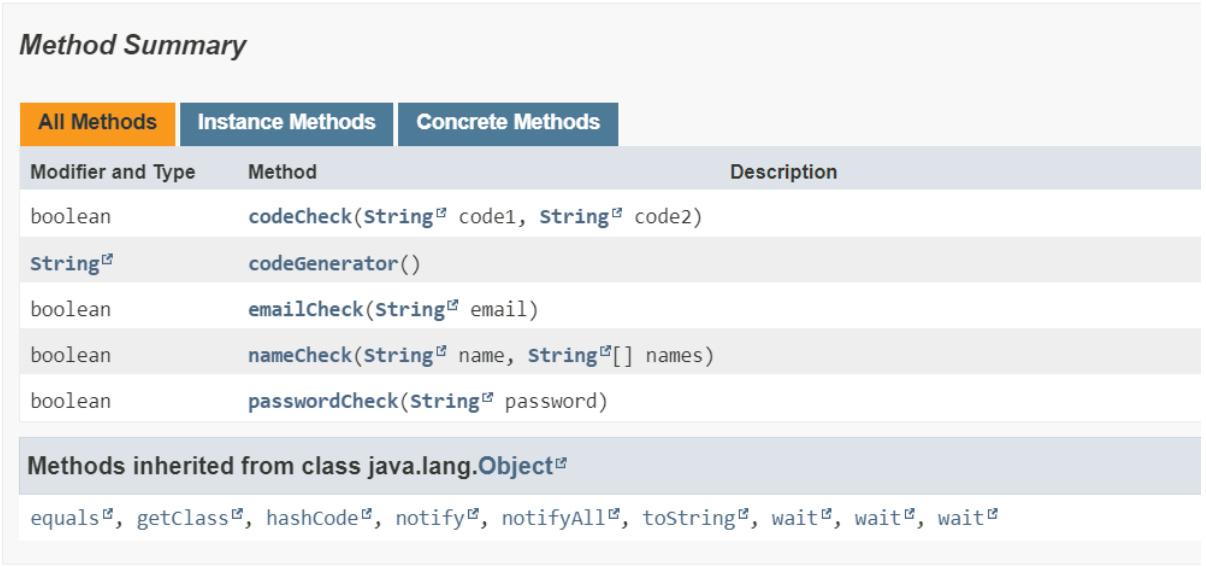
Modifier and Type	Method	Description
static void	`codeInput()`	Prompts the user to input the code.
static void	`emailInput()`	Prompts the user to input their email.
static void	`main(String[] args)`	Prompts the user to input their name, email, password, and code, and calls the check method from the validarCampos class to validate the inputs.
static void	`nameInput()`	Prompts the user to input their name.
static void	`passwordInput()`	Prompts the user to input their password.



Constructor Summary

Constructors

Constructor	Description
`validarCamposNuevo()`	MÃ©todo que comprueba si los campos introducidos son vÃ¡lidos y el cÃ³digo de segurida



Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
boolean	`codeCheck(String code1, String code2)`	
String	`codeGenerator()`	
boolean	`emailCheck(String email)`	
boolean	`nameCheck(String name, String[] names)`	
boolean	`passwordCheck(String password)`	

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Link al github: https://github.com/rikipove/practicaEntorns