

# Software Distribuït

## Pràctica 0: Bytes i Streams

# Presentació Professorat

**Eloi Puertas, Blai Ras, David Martínez Carpena**

**Contacte:**

- Campus Virtual
- GitHub

# Presentació Sessions de Pràctiques

**S'usarà Campus Virtual i Github Classroom.**

Pràctiques que es realitzaran durant el quadrimestre:

**Projecte 1:** Sockets Client/Servidor (Java)

**Projecte 2:** Desenvolupament Web/API Rest (Flask i Vue.js )

# Grups de Pràctiques

Les pràctiques es realitzaran en grups de **2 persones**.

Farem servir github classroom per crear els repositoris i els grups (Teams)

Farem servir CodeReview per puntuar les pràctiques. La nota del codi serà 50% pel desenvolupament de codi i 50% per la revisió

**Qualsevol problema que tingueu amb el funcionament de la parella de pràctiques contacteu el més aviat possible amb el professor/a. Cada membre de la parella haurà de desenvolupar i revisar codi.**

# Pràctica 0

Els principals objectius d'aquesta sessió:

- Entendre i saber utilitzar les funcions que formen part de **ComUtils.java**
- Saber llegir i escriure correctament diferents tipus de dades
- Entendre la diferència entre les funcions i en quin cas utilitzar cadascuna d'elles.
- Ser capaç de modificar aquestes funcions de base per aconseguir escriure i llegir un missatge amb unes característiques concretes.
- Comprendre el test d'exemple i realitzar-ne de nous.
- Saber executar el codi i els tests desde l'editor i/o consola.

**IMPRESINDIBLE PER PODER FER LA PRÀCTICA 1**

# Pràctica 0

Podeu trobar l'enunciat de la Pràctica 0 i el codi corresponent al assignament del classroom [https://classroom.github.com/a/E\\_rfSPVE](https://classroom.github.com/a/E_rfSPVE)

El podeu obrir en qualsevol editor de text però està preparat per ser obert com a un Projecte de IntelliJ

# Pràctica 0

Se us demana fer una sèrie de modificacions per estendre les classes de la llibreria **utils** (**ComUtils.java** i **ComUtilsService.java**) , que s'utilitzarà i ampliarà al llarg del Projecte 1.

En aquesta sessió emularem el **DataInputStream** i **DataOutputStream** que utilitzarem més endavant en la comunicació entre sockets amb els que s'utilitzen per escriure i llegir per fitxer. D'aquesta manera podrem visualitzar i quantificar els bytes que s'escriuen/llegeixen.

# Pràctica 0

Les funcions principals que componen **ComUtils.java** són:

- **String: read\_string()**
- **int: read\_int32()**
- **String: read\_string\_variable(int head\_size)**
  
- **write\_string(String s)**
- **write\_int32(int)**
- **write\_string\_variable(int head\_size, String s)**

Les funcions estan explicades amb més detall en l'enunciat corresponent.



# Pràctica 0

## EXERCICIS EXTRES PROPOSATS:

Realitzar l'escriptura/lectura d'un fitxer amb el següent format:

**String[20]** <El vostre nom i cognom>

**int32** <Edat>

**String[]** <Comentari>

- Que observeu? Quants bytes s'han escrit en el fitxer? Tots són llegibles?
- En quin ordre s'han de llegir? Quines funcions heu fet servir?

# Pràctica 0

Al final d'aquesta sessió hauríeu de ser capaços de respondre a les següents preguntes:

- Quina diferència hi ha entre **read\_string** i **read\_string\_variable**? Són intercanviables?
- Si volem que un int32 sigui llegible en un fitxer, que haurem de fer?
- Quines creus que poden ser altres funcions útils?

# Pràctica 0

```
public class ComUtilsTest {  
    @Test  
    public void example_test() {  
        File file = new File("test");  
        try {  
            file.createNewFile();  
            ComUtils comUtils = new ComUtils(new FileInputStream(file),  
            new FileOutputStream(file));  
  
            comUtils.write_int32(2);  
            int readedInt = comUtils.read_int32();  
            assertEquals(2, readedInt);  
  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

# Pràctica 0

Serieu capaços de realitzar les funcions **write\_char** i **read\_char** i realitzar-ne els tests corresponents?

Aquí un link que us pot ser útil: <https://www.guru99.com/junit-assert.html>