

## **FAMÍLIA PROFESSIONAL: INFORMÀTICA i COMUNICACIONS**

**CICLE FORMATIU: CFPS ICC0**

**CFGS Desenvolupament d'aplicacions web**

**Curs acadèmic: 23-24**

### **PROGRAMACIÓ PROJECTE DE CICLE:**

**Global Project**

### **NOM DEL PROJECTE:**

**Frameworks avançats de JS**

### **RÚBRICA DE LA REALITZACIÓ DE LA FASE 4 PART 1**

Criteris	4	3	2	1
Sistema d'autenticació (10%)	El sistema de login està implementat i hi ha logout. Existeix el rol d'administrador,	El sistema de login està implementat i hi ha logout.	El sistema de login està implementat però no hi ha logout.	El sistema de login no està correctament implementat
Lògica de l'aplicació en NodeJS (35%)	La lògica és correcta i funcional. Es fa ús d'encaminament.	La lògica és correcta i funcional. Hi ha una bona organització modular. No s'utilitza encaminament.	La lògica és correcta i funcional. No s'utilitza encaminament.	La lògica és incorrecta i l'aplicació no és funcional

POO del frontend (10%)	S'ha creat diverses classes però amb jerarquia entre elles. Les classes estan implementades només en ES6 i també fan ús de programació modular en el frontend.	S'ha creat diverses classes però amb jerarquia entre elles. Les classes estan implementades només en ES6.	S'ha creat diverses classes però sense jerarquia entre elles.	Només s'ha creat una classe.
HTML5 (30%)	Es fa servir les API d'emmagatzge web, canvas i D&D	Es fa servir les API d'emmagatzge web, canvas i D&D	Es fa servir canvas i D&D	Només es fa servir l'API de canvas de HTML5
MongoDB (15%)	S'ha utilitzat persistència de dades amb MongoDB amb totes les operacions CRUD implementades i es fa ús del patró DAO.	S'ha utilitzat persistència de dades amb MongoDB amb totes les operacions CRUD implementades però sense utilitzar el patró DAO.	S'ha utilitzat persistència de dades amb MongoDB, però és molt bàsica.	No s'ha utilitzat persistència de dades. Només es llegeixen dades des del MongoDB.

## RÚBRICA DE LA REALITZACIÓ DE LA FASE 4 PART 2

Criteris	4	3	2	1
Sistema d'autenticació (10%)	El sistema de login està implementat i hi ha logout. Existeix el rol d'administrador. Es fa servir Pug en totes les interfícies	El sistema de login està implementat i hi ha logout. Es fa servir Pug en totes les interfícies	El sistema de login està implementat però no hi ha logout.	El sistema de login no està correctament implementat
Lògica de l'aplicació en Express ( 35%)	La lògica és correcta. S'indica visualment clarament tota la informació relativa a l'escassetat de l'aigua. S'utilitza encaminament d'ExpressJS. L'aplicació utilitza el protocol HTTPS.	La lògica és correcta. S'indica visualment clarament tota la informació relativa a l'escassetat de l'aigua. S'utilitza encaminament d'ExpressJS.	La lògica és correcta però molt bàsica. No s'utilitza encaminament.	La lògica és incorrecta i l'aplicació no és funcional
Geolocalització (10%)	S'ha implementat una Geolocalització molt bàsica. Es visualitza en un Mapa (OpenStreetMaps).			S'ha implementat una Geolocalització molt bàsica. Només surt per consola.
POO del backend (20%)	S'ha creat diverses classes però amb jerarquia entre elles. Les classes estan implementades només en ES6 i	S'ha creat diverses classes però amb jerarquia entre elles. Les classes estan	S'ha creat diverses classes però sense jerarquia entre elles.	Només s'ha creat una classe.

	també fan ús de programació modular en el frontend.	implementades només en ES6.		
MongooSE (25%)	S'ha utilitzat persistència de dades amb MongoDB amb totes les operacions CRUD implementades i els esquemes estan molt ben definits.	S'ha utilitzat persistència de dades amb MongoDB amb totes les operacions CRUD implementades però els esquemes són molt bàsics.	S'ha utilitzat persistència de dades amb MongoDB, però és molt bàsica.	No s'ha utilitzat persistència de dades. Només es llegeixen dades des del MongoDB.