

CONCURS LABORAL INTERN TORN DE PROMOCIÓ PER COBRIR TRES LLOCS DE TREBALL DE TÈCNIC/A SUPERIOR EN INFORMÀTICA I COMUNICACIONS AMB PERFIL DE TÈCNIC/A EN IC DE NIVELL 1 PER A DIFERENTS CAMPUS DE LA UPC. CODIS 184-50 I 195-137.

PROVA DE CONEIXEMENTS PRÀCTICS

La prova consta de diverses preguntes que cal respondre breument.

Aquesta prova avaluarà les teves habilitats pràctiques i teòriques, centrant-se en la teva capacitat per resoldre problemes i executar tasques crítiques en l'entorn TIC de la UPC.

Puntuació màxima: 4 punts

Per calcular el resultat final de la prova es dividirà el total de punts obtinguts per 20 i es multiplicarà per 4.

Temps: 2 hores

CONTEXT:

La teva cap TIC de la UTG t'ha proposat un nou projecte per desenvolupar i posar en marxa un sistema d'informació per a la gestió d'equipaments IoT (Internet of Things) pel suport a la monitorització i manteniment de les instal·lacions del campus.

Aquest sistema ha de donar servei principalment a:

- El personal de Manteniment del Campus (tècnics que s'encarreguen del manteniment correctiu i preventiu dels equipaments).
- Els responsables de Sostenibilitat i Seguretat de la universitat, que utilitzen les dades ambientals i de videovigilància.

El teu equip de treball estarà format per un/a altre desenvolupador/a, un/a becari/a i es treballarà amb les eines definides al Marc de desenvolupament de la UPC.

Tot i que el sistema està en fase inicial i els requisits encara s'han d'acabar de definir, es preveu que ha de permetre:

1. Inventariar i gestionar els sensors, càmeres i dispositius IoT del campus (ubicació, tipus, estat, data d'instal·lació, garanties, contractes de manteniment, etc.).
2. Alertar de manera automàtica quan es detectin anomalies.
3. Registrar intervencions de manteniment i canvis.
4. Emmagatzemar i gestionar dades recollides pels dispositius, com ara imatges, dades ambientals, vídeos, accessos o registres de consum.
5. Facilitar la consulta de dades per part d'altres sistemes, especialment aquells relacionats amb sostenibilitat, seguretat o sistemes d'alerta.
6. Garantir la seguretat de l'accés i els permisos, de manera que només el personal autoritzat pugui fer canvis, i es mantingui un registre d'activitat (auditoria).
7. Complir amb els criteris de seguretat i protecció de dades exigits per la normativa de la UPC, el RGPD i l'ENS (Esquema Nacional de Seguretat).

Pregunta 1: Gestió de Projectes (5 punts)

1.1. Fases del Projecte (2 punts)

Descriu breument (màxim 1 pàgina) les fases que seguiries fins a arribar a posar el sistema en producció.

1.2. Eines de Planificació (1 punt)

Esmenta almenys 2 eines que podries fer servir per a la planificació del projecte i 2 per al repositori de documentació i 1 per al control de versions. Quines escolliries a la UPC i explica en un paràgraf per què?

1.3. Metodologia Agile (2 punts)

Tenint en compte que es farà servir una metodologia AGILE, respon a les següents preguntes (0,25 punts per pregunta, les respostes errònies descompten 0,07 punts):

- 1. Quina figura és l'encarregada de prioritzar el Product Backlog?**
 - a) Scrum Master
 - b) Product Owner
 - c) Stakeholder
 - d) Cap de les anteriors és correcta
- 2. Quina figura té l'última paraula per a valorar l'esforç d'una història d'usuari?**
 - a) Product Owner
 - b) Developer
 - c) Stakeholder
 - d) Cap de les anteriors és correcta
- 3. Quant dura un sprint?**
 - a) entre una setmana i quatre setmanes
 - b) 1 dia
 - c) varia segons la fase del projecte
 - d) Totes les anteriors són correctes
- 4. Al final de cada sprint se sol fer:**
 - a) Sprint planning
 - b) Sprint review i Sprint retrospective
 - c) Daily scrum
 - d) Final scrum review
- 5. Quina d'aquestes afirmacions descriu millor una metodologia Agile?**
 - a) Es basa en la planificació detallada de totes les fases del projecte abans de començar.
 - b) Es treballa en iteracions curtes per adaptar-se millor als canvis dels requisits.
 - c) Es fan entregues úniques al final del projecte, un cop tot està acabat.
 - d) El client no participa durant el desenvolupament, només al final.
- 6. Com s'anomena el període de temps fix durant el qual es desenvolupen funcionalitats en Scrum?**
 - a) Iteració final
 - b) Roadmap
 - c) Sprint
 - d) Kick-off
- 7. En un projecte Agile, quin d'aquests rols s'encarrega de definir les funcionalitats que aporten valor al client?**
 - a) Scrum Master
 - b) Equip de desenvolupament
 - c) Product Owner
 - d) Cap de sistemes
- 8. Quin dels següents documents o artefactes és típic d'un projecte Agile?**
 - a) Gantt chart
 - b) Backlog del producte
 - c) Diagrama PERT
 - d) Especificació funcional tancada

Indica les teves respostes en la següent graella:

1	2	3	4	5	6	7	8

Pregunta 2: Disseny d'Arquitectura (5 punts)

2.1. Components del Sistema (2 punts)

Tal i com t'han demanat, has de fer un disseny d'una aplicació que tindrà una arquitectura Single Page Application/REST.

Indica quins són els principals components del sistema de la teva aplicació (que tindrà una arquitectura Single Page Application/REST) i què fan de forma general

(màxim 1 pàgina en total).

2.2. Connexió de Components (1,5 punts)

Amb els components principals definits a la pregunta anterior, com es connectarien entre ells, amb quin protocol i format de dades?

Omple aquesta taula amb les línies que necessitis per a cada component:

Component origen	Component destí	Protocol	Format dades
...			

2.3. Tecnologies (1,5 punts)

Segons el marc de desenvolupament de la UPC, quina tecnologia i programari faries servir per a cadascun dels components?

Omple aquesta taula amb les línies que necessitis per a cada component:

Tecnologia Proposada per Component (Segons Marc UPC):

Component	Tecnologia proposta
...	

Pregunta 3: Desenvolupament (6 punts)

3.1. Consulta Django ORM (1,5 punts)

Tens el següent model Django simplificat:

```
Python
from django.db import models

class Sensor(models.Model):
    codi = models.CharField(max_length=100)
    tipus = models.CharField(max_length=50)
    ubicacio = models.CharField(max_length=100)

class Lectura(models.Model):
    sensor = models.ForeignKey(Sensor, on_delete=models.CASCADE,
related_name='lectures')
    valor = models.FloatField()
    data_hora = models.DateTimeField()
```

Vols recuperar totes les lectures dels sensors de tipus "temperatura" o ubicats a "Aula 101", carregant la informació del sensor de forma eficient.

Quin dels següents fragments de codi és el més adequat i per què?

a)

```
Python
Lectura.objects.filter(sensor__tipus="temperatura" or
sensor__ubicacio="Aula 101")
```

b)

```
Python
Lectura.objects.filter(Q(sensor__tipus="temperatura"),
Q(sensor__ubicacio="Aula 101")).select_related('sensor')
```

c)

```
Python
Lectura.objects.filter(
    Q(sensor__tipus="temperatura") | Q(sensor__ubicacio="Aula 101")
).select_related('sensor')
```

d)

```
Python
Lectura.objects.select_related('sensor').filter(
    sensor__tipus="temperatura", sensor__ubicacio="Aula 101"
)
```

3.2. Renderització Condicional a Vue 3 (1,5 punts)

Vols mostrar dades obtingudes d'una API amb Axios dins d'un component Vue 3, i vols assegurar-te que la taula d'Element Plus només es mostri quan ja s'ha completat la petició i es tenen dades, evitant mostrar informació buida o incorrecta.

Quin dels següents fragments de codi fa això correctament i per què?

a)

HTML

```
<template>
  <el-table v-if="!loading" :data="sensors">
    <el-table-column prop="nom" label="Nom" />
    <el-table-column prop="valor" label="Valor" />
  </el-table>
  <div v-else>Carregant dades...</div>
</template>

<script>
import axios from 'axios'

export default {
  data() {
    return {
      sensors: [],
      loading: true,
    }
  },
  mounted() {
    axios.get('/api/sensors/data')
      .then(response => {
        this.sensors = response.data
        this.loading = false
      })
  }
}
</script>
```

b)

HTML

```
<template>
  <el-table v-if="loading" :data="sensors">
    <el-table-column prop="nom" label="Nom" />
    <el-table-column prop="valor" label="Valor" />
  </el-table>
  <div v-else>Carregant dades...</div>
</template>

<script>
import axios from 'axios'

export default {
  data() {
    return {
      sensors: [],
      loading: true,
    }
  },
  mounted() {
    axios.get('/api/sensors/data')
      .then(response => {
        this.sensors = data
        this.loading = false
      })
  }
}
```

```
}  
</script>
```

c)

HTML

```
<template>  
  <el-table v-if="loading" :data="sensors">  
    <el-table-column prop="nom" label="Nom" />  
    <el-table-column prop="valor" label="Valor" />  
  </el-table>  
  <div v-else>Carregant dades...</div>  
</template>
```

```
<script>  
import axios from 'axios'  
  
export default {  
  data() {  
    return {  
      sensors: [],  
      loading: true,  
    }  
  },  
  mounted() {  
    axios.get('/api/sensors/data')  
      .then(response => {  
        sensors = response.data  
        loading = false  
      })  
  }  
}  
</script>
```

d)

HTML

```
<template>  
  <el-table v-if="this.loading" :data="sensors">  
    <el-table-column prop="nom" label="Nom" />  
    <el-table-column prop="valor" label="Valor" />  
  </el-table>  
  <div v-else>Carregant dades...</div>  
</template>
```

```
<script>  
import axios from 'axios'  
  
export default {  
  data() {  
    return {  
      sensors: [],  
      loading: true,  
    }  
  },  
  mounted() {  
    axios.get('/api/sensors/data')  
      .then(response => {  
        this.sensors = response.data  
      })  
  }  
}
```

```
        this.loading = false
    })
}
}
</script>
```

3.3. Consum d'API REST amb Python (3 punts)

Vull consultar l'API REST que em proporciona un sistema de sensors propietari, ubicada a la URL: <https://iotmarcaxx.upc.edu>.

La documentació de l'API indica:

- **Endpoint:** `/api/sensors/`
- **Mètode:** `GET`
- **Paràmetres:**
 - **tipus** (ex: temperatura, humitat, imatges)
 - **ubicacio** (ex: Aula101, Aula102, Aula103, SalaJunes)
- **Resposta (format JSON):**

JSON

```
[
  {
    "id": 12,
    "codi": "TMP-A101-1",
    "tipus": "temperatura",
    "ubicacio": "Aula101",
    "darrera_lectura": {
      "valor": 23.5,
      "data_hora": "2025-06-17T12:35:00Z"
    }
  },
  ...
]
```

Escriu el codi Python, utilitzant la llibreria `requests`, necessari per obtenir i imprimir els valors de temperatura dels sensors de l'Aula101. (màxim 1/2 pàgina)

Pregunta 4: Seguretat i Protecció de Dades (4 punts)

4.1. Aspectes de Seguretat en Aplicacions Web (3 punts)

Quins aspectes tindràs en compte durant el desenvolupament de l'aplicació web per garantir que sigui segura? Detalla les principals mesures que aplicaràs tant al *frontend* com al *backend*. (1 pàgina)

4.2. Tractament de Dades Sensibles (1 punt)

Consideres que hi ha algun tipus de dades que requerirà un tractament especial per motius de protecció de dades personals segons el RGPD i l'ENS? Justifica la resposta (màxim 1 pàgina)