

**Exame Modelo de Sistemas Distribuídos
2021-22 – 2º Semestre**

LEI e LECD
2h

AVISOS

1 – A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro Mestre. Qualquer tentativa de fraude leva à anulação da prova tanto do facilitador como do prevaricador.

2 – Este teste é individual e com consulta. Não pode usar dispositivos eletrónicos, incluindo o telemóvel. Não pode trocar apontamentos com colegas.

NOME: _____ NÚMERO: _____

- 1- Uma *thread* num `wait()` dum monitor impede a entrada de outra *thread* no monitor?
Justifique brevemente.

- 2- Indique, justificando, se o protocolo UDP seria adequado para trocar ficheiros entre um cliente e um servidor.

- 3- Explique em traços gerais como é que a Bitcoin impede o duplo gasto.



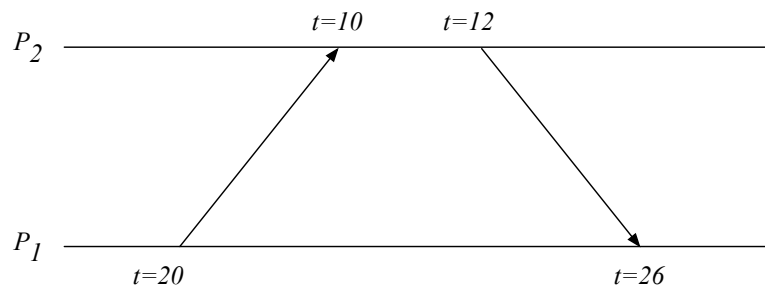
- 4- Que problema(s) é que o modelo RRA (Request-Reply-Acknowledge) poderá levantar na implementação de um sistema de RPCs?

- 5- Indique se uma *cache* HTTP poderá armazenar o resultado de um POST e servir esse resultado ao cliente num próximo pedido? Justifique.

- 6- Indique os principais passos envolvidos na assinatura digital de um documento e respetiva verificação.

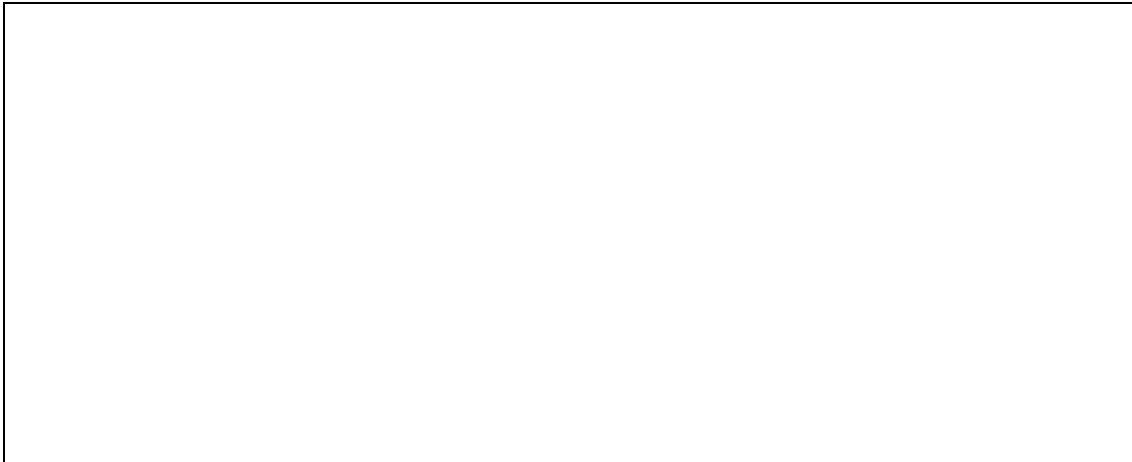
- 7- Que limitações poderia ter a utilização duma tabela de dispersão distribuída (DHT) como o Chord como serviço de nomes da Internet (i.e., para substituir o DNS)?

- 8- (só para a LEI) Considere a seguinte figura que ilustra o processo de sincronização de relógios no algoritmo NTP simétrico:



Qual é o acerto que P_1 tem de fazer relativamente a P_2 (*offset*)? Assinale a alínea correta (assinalar uma opção errada desconta cotação_da_pergunta/(número_de_alternativas - 1)):

- a) -12
 - b) -10
 - c) 2
 - d) -2
 - e) Nenhuma das anteriores.
- 9- (possível alternativa para a LECD) Para que servem os web sockets?



10- Quantas tabelas na base de dados resultam da utilização destas entidades no Java Persistence API? Justifique descrevendo sucintamente o seu formato.

```
package jpaprimer;

import java.util.List;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;

@Entity
public class Professor {
    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String name, office;
    @OneToMany(mappedBy = "prof")
    private List<Student> st;

    public Professor() {}

    public Professor(String name, String office) {
        this.name = name;
        this.office = office;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

```

    }

    public String getName() {
        return name;
    }

    public String getOffice() {
        return office;
    }

    public void setOffice(String office) {
        this.office = office;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String toString() {
        return this.name + ". Office: " + this.office;
    }

    public List<Student> getStudents() {
        return this.st;
    }
}

```

```

package jpaprimer;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

@Entity
public class Student {
    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String name, telephone;
    private int age;
    @ManyToOne(cascade = CascadeType.ALL)
    private Professor prof;

    public Student() {
    }
}

```

```
public Student(String name, String telephone, int age, Professor prof) {  
    this.name = name;  
    this.telephone = telephone;  
    this.age = age;  
    this.prof = prof;  
}  
  
public String getTelephone() {  
    return telephone;  
}  
  
public Professor getProf() {  
    return prof;  
}  
  
public void setProf(Professor prof) {  
    this.prof = prof;  
}  
  
public int getAge() {  
    return age;  
}  
  
public void setAge(int age) {  
    this.age = age;  
}  
  
public String getName() {  
    return name;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
  
public void setTelephone(String telephone) {  
    this.telephone = telephone;  
}  
  
public String toString() {  
    return this.name + ". Telephone: " + this.telephone + ". Age: " +  
this.age;  
}  
}
```

