# Prova 1

## Domanda 1

Una P.A. utilizza applicativi gestionali in modalità SAAS. Questi NON espongono API sui dati, ma consentono un accesso read-only ai DB di back-end tramite VPN "punto a punto".
Un outsourcer della stessa P.A. chiede ai Servizi IT la disponibilità di alcune tipologie di dati da questi applicativi per integrarli in un nuovo portale.
1) Si definisca una possibile soluzione applicativa per soddisfare tale richiesta nel rispetto del paradigma "zero trust".
2) Si analizzi i possibili scenari per l'autenticazione dell'integrazione da proporre all'outsourcer.
3) Si valuti altresì la richiesta dell'outsourcer nei termini della vigente normativa sulla privacy.

## Domanda 2

Si consideri il seguente frammento di codice PHP:

```
<?php
$conn = mysqli_connect("localhost", "root", "password", "mydb");

$username = $_GET['username'];
$password = $_GET['password'];

$query = "SELECT * FROM users WHERE username = '$username' AND password = '$password'";
$result = mysqli_query($conn, $query);

if(mysqli_num_rows($result) > 0){
   echo "Login effettuato!";

   setcookie("user", $username, time()+3600);

   if($_GET['redirect']){
      header("Location: " . $_GET['redirect']);
   }
} else {
   echo "Credenziali non valide";
}

if(isset($_GET['page'])){
   include($_GET['page'] . ".php");
}
?>
```

Il candidato:

- Individui e descriva le principali vulnerabilità di sicurezza presenti nel codice.
- Classifichi le tipologie di vulnerabilità (es. secondo OWASP Top 10 o categorie analoghe).
- Spieghi le possibili conseguenze di tali vulnerabilità in un contesto reale.

Proponga soluzioni tecniche concrete per mitigare o risolvere i problemi individuati, motivando le scelte (es. tecniche di hardening, validazione input, gestione autenticazione, configurazione server, ecc.).

La risposta deve essere strutturata e motivata, con riferimento ai principi di sicurezza applicativa (secure coding, least privilege, defense in depth).

## Domanda 3

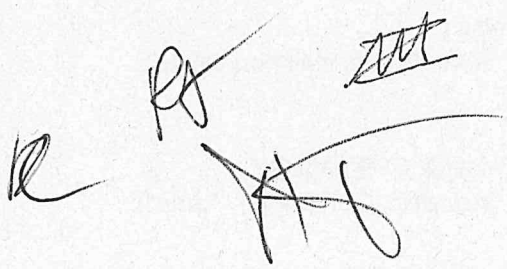
Un ente deve realizzare una nuova piattaforma applicativa per la gestione di dati sensibili, composta da:

- Frontend web accessibile da Internet
- API backend REST
- Servizi interni di elaborazione asincrona
- Database relazionale
- Sistema di autenticazione centralizzato (es. LDAP/OIDC)
- Integrazione con servizi esterni (es. pagoPA, sistemi regionali)

L'infrastruttura è ospitata in un data center ibrido (parte on-premise, parte cloud) e deve rispettare i principi di: Segregazione delle reti (VLAN / subnet dedicate), Zero Trust, Least privilege, Alta disponibilità, Minimizzazione della superficie di attacco.

Si chiede al candidato di descrivere:

- Come gestirebbe logging, monitoring e troubleshooting in un contesto con load balancer e segmentazione multi-layer.
- Quali errori progettuali tipici commette uno sviluppatore che non ha competenze di networking e quali conseguenze possono derivarne.



# Prova 2

## Domanda 1

Un ente pubblico utilizza un sistema informativo monolitico legacy sviluppato in PHP, privo di test automatizzati, con architettura fortemente accoppiata e integrazioni multiple verso sistemi esterni (autenticazione, pagamenti, protocollo, database relazionali). Il sistema supporta processi critici e non può essere interrotto.

L'amministrazione intende evolvere o sostituire il sistema.

Il candidato descriva:

1. Le principali criticità architetturali tipiche di un monolite legacy e le implicazioni sulla manutenibilità, sicurezza e scalabilità.
2. Le problematiche tecniche legate alla migrazione incrementale delle funzionalità e dei dati verso una nuova architettura.
3. I possibili pattern di migrazione applicabili (es. Strangler Pattern, modularizzazione progressiva, API façade, replatforming, rehosting), illustrandone vantaggi e limiti.
4. I criteri tecnici per decidere tra:
    - riscrittura completa,
    - rifattorizzazione progressiva,
    - adozione di una soluzione di mercato.
5. Le misure che adotterebbe per garantire qualità, continuità del servizio e controllo del rischio durante la transizione.

Si richiede una risposta che evidenzi competenze di progettazione software, gestione del debito tecnico e modernizzazione applicativa.

## Domanda 2

Si consideri il seguente frammento di codice **Python (FastAPI)**:

```
from fastapi import FastAPI, Request
from fastapi.responses import RedirectResponse, PlainTextResponse
import json, os, requests
app = FastAPI()
# Finto archivio utenti (in un contesto reale sarebbe un DB/servizio)
USERS = {
    "admin": {"password": "admin123", "role": "admin"},
    "mario": {"password": "pwd", "role": "user"}
}
@app.get("/auth")
async def auth(request: Request):
    u = request.query_params.get("u", "")
```

```
    p = request.query_params.get("p", "")

    if u in USERS and USERS[u]["password"] == p:
        token = json.dumps({"user": u, "role": USERS[u]["role"]})
        nxt = request.query_params.get("next")
        resp = PlainTextResponse("OK")
        resp.set_cookie("session", token)
        if nxt:
            return RedirectResponse(nxt)
        return resp
    return PlainTextResponse("NO", status_code=401)
@app.get("/fetch")
async def fetch(url: str):
    r = requests.get(url, timeout=3)
    return PlainTextResponse(r.text[:5000])
@app.get("/download")
async def download(file: str):
    path = os.path.join("/var/app/uploads", file)
    with open(path, "rb") as f:
        return PlainTextResponse(f.read().decode("utf-8", errors="ignore")
[:5000])
```

Il candidato:

- individui e descriva le vulnerabilità principali;
- le classifichi (OWASP o categorie analoghe);
- discuta gli impatti realistici (es. compromissione account, escalation, esfiltrazione dati, pivot interno, phishing);
- proponga mitigazioni concrete: validazione input, gestione sessione/token, restrizioni di rete/egress, allowlist, hardening, logging/monitoring, ecc.

## Domanda 3

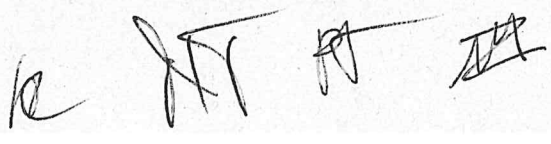
Un ente deve realizzare una nuova piattaforma applicativa per la gestione di dati sensibili, composta da:

- Frontend web accessibile da Internet
- API backend REST
- Servizi interni di elaborazione asincrona
- Database relazionale
- Sistema di autenticazione centralizzato (es. LDAP/OIDC)
- Integrazione con servizi esterni (es. pagoPA, sistemi regionali)

L'infrastruttura è ospitata in un data center ibrido (parte on-premise, parte cloud) e deve rispettare i principi di: Segregazione delle reti (VLAN / subnet dedicate), Zero Trust, Least privilege, Alta disponibilità, Minimizzazione della superficie di attacco.

Si chiede al candidato di descrivere:

- Come progettarebbe la segmentazione della rete (es. DMZ, rete applicativa, rete di management), motivando le scelte.
- Quali implicazioni tale segmentazione ha sul design dell'applicazione (es. gestione delle configurazioni, discovery dei servizi, gestione degli errori di rete, timeouts, resilienza).



# Prova 3

## Domanda 1

A seguito di un accorpamento tra Enti i 2 team di sviluppo ricevono il compito di integrare i rispettivi applicativi di gestione delle presenze.

1) Si definisca in modo schematico le fasi di questa attività in modo da soddisfare i requisiti di entrambe le Amministrazioni, nel rispetto delle linee guida AGID.

2) Si riassuma le modalità di condivisione del codice tra i 2 team.

3) Si indichi i possibili scenari di rilascio delle nuove funzionalità comuni.

## Domanda 2

Si consideri il seguente frammento di codice **Python (Flask)**:

```
from flask import Flask, request, redirect, make_response
import sqlite3
app = Flask(__name__)
def db():
    return sqlite3.connect("app.db")
@app.get("/login")
def login():
    username = request.args.get("username", "")
    password = request.args.get("password", "")

    q = f"SELECT id, username FROM users WHERE username='{username}' AND
password='{password}'"
    conn = db()
    cur = conn.cursor()
    cur.execute(q)
    row = cur.fetchone()
    if row:
        resp = make_response("Login effettuato!")

        resp.set_cookie("user", username, max_age=3600)

        redir = request.args.get("redirect")
        if redir:
            return redirect(redir)
        return resp
    return "Credenziali non valide", 401
@app.get("/page")
def page():
    name = request.args.get("name")

    if name:
        with open(f"pages/{name}.html", "r", encoding="utf-8") as f:
            return f.read()
    return "missing name", 400
```

Il candidato:

- individui e descriva le principali vulnerabilità presenti;
- le classifichi (es. OWASP Top 10 o categorie equivalenti);
- spieghi le conseguenze in un contesto reale;
- proponga soluzioni tecniche concrete (secure coding, least privilege, defense in depth), motivandole.

## Domanda 3

Un ente deve realizzare una nuova piattaforma applicativa per la gestione di dati sensibili, composta da:

- Frontend web accessibile da Internet
- API backend REST
- Servizi interni di elaborazione asincrona
- Database relazionale
- Sistema di autenticazione centralizzato (es. LDAP/OIDC)
- Integrazione con servizi esterni (es. pagoPA, sistemi regionali)

L'infrastruttura è ospitata in un data center ibrido (parte on-premise, parte cloud) e deve rispettare i principi di: Segregazione delle reti (VLAN / subnet dedicate), Zero Trust, Least privilege, Alta disponibilità, Minimizzazione della superficie di attacco.

Si chiede al candidato di descrivere:

- Come implementerebbe, i principi di: Secure communication (TLS, mutual TLS, certificate validation), Gestione sicura delle credenziali, Autorizzazione tra servizi (service-to-service authentication).
- Quali pattern architetturali adotterebbe per rendere il sistema coerente con la segmentazione di rete (es. API Gateway, Reverse Proxy, Service Mesh, Circuit Breaker, Dependency Injection).