

## DESARROLLAR UN PROGRAMA EXPLORADOR DE PHISHING EN JAVA

### 1.- Investigar estafas de phishing en línea

Las estafas de phishing en línea son intentos fraudulentos de obtener información personal y confidencial, como contraseñas, números de tarjetas de crédito o información bancaria, haciéndose pasar por una entidad de confianza. Estos estafadores suelen enviar correos electrónicos, mensajes de texto o mensajes en redes sociales que parecen legítimos, pero en realidad están diseñados para engañar a las personas y hacer que revelen su información.

medidas de seguridad para protegerte contra las estafas de phishing

1.Desconfiar de correos electrónicos, mensajes o enlaces sospechosos, especialmente si solicitan información confidencial.

2.Verificar la autenticidad de los sitios web ingresando la dirección directamente en el navegador.

3.Mantener actualizados los programas de seguridad en tu computadora y dispositivos móviles.

4.No proporcionar información personal o financiera a menos que estés seguro de la legitimidad de la fuente.

5.Estar atento a errores gramaticales o de ortografía en los mensajes sospechosos, ya que a menudo son indicios de una estafa.

6.Utilizar contraseñas fuertes y únicas para tus cuentas en línea

2. Con esa información, Usted debe crear una lista de 30 palabras, frases y nombres de compañías que se encuentren frecuentemente en los mensajes de phishing y asignarle el valor de un punto a cada una, con base en una estimación de la probabilidad de que aparezca en un mensaje de phishing (por ejemplo, un punto si es poco probable, dos puntos si es algo probable, o tres puntos si es muy probable).

1. "Urgente" - 2 puntos
2. "Verificación de cuenta" - 2 puntos
3. "Actualice su información" - 2 puntos
4. "Gane dinero rapido" - 3 puntos
5. "Premio en efectivo" - 2 puntos
6. "Problema de seguridad" - 2 puntos
7. "Contraseña" - 1 punto
8. "Cuenta suspendida" - 2 puntos
9. "Tarjeta de crédito" - 1 punto
10. "Gane un viaje gratis" - 2 puntos
11. "Confirme su identidad" - 2 puntos
12. "Oferta limitada" - 1 punto
13. "Iniciar sesión para reclamar" - 2 puntos
14. "Pago pendiente" - 1 punto
15. "Actualización de software" - 1 punto

15. "Problema con su pedido" - 1 punto
16. "Alerta de seguridad" - 2 puntos
17. "Verificación de seguridad" - 2 puntos
18. "Restricción de cuenta" - 2 puntos
19. "Banco" - 1 punto
20. "Tarjeta bancaria" - 1 punto
21. "Phishing" - 3 puntos
22. "Premio de lotería" - 2 puntos
23. "Gane un iPhone" - 2 puntos
25. "Actualizar sus datos" - 1 punto
26. "Problema con su cuenta" - 2 puntos
27. "Gane dinero desde casa" - 3 puntos
28. "Oferta exclusiva" - 1 punto
29. "Restablecer contraseña" - 2 puntos
30. "Gane un millón de dólares" - 3 puntos

3. Escriba una aplicación que explore un archivo de texto en busca de estos términos y frases en java.

```
public class PhishingDetector {
    public static void main(String[] args) {
        String filePath = "ruta/al/archivo.txt"; // Ruta del archivo de
        texto a explorar

        // Palabras, frases y nombres de compañías a buscar
        String[] phishingTerms = {
            "Urgente", "Verificación de cuenta", "Actualice su
información", "Gane dinero rápido",
            "Premio en efectivo", "Problema de seguridad",
            "Contraseña", "Cuenta suspendida",
            "Tarjeta de crédito", "Gane un viaje gratis", "Confirme
su identidad", "Oferta limitada",
            "Inicie sesión para reclamar", "Pago pendiente",
            "Actualización de software",
            "Problema con su pedido", "Alerta de seguridad",
            "Verificación de seguridad",
            "Restricción de cuenta", "Banco", "Tarjeta bancaria",
            "Phishing", "Premio de lotería",
            "Gane un iPhone", "Actualizar sus datos", "Problema con
su cuenta", "Gane dinero desde casa",
            "Oferta exclusiva", "Restablecer contraseña", "Gane un
millón de dólares"
        };

        int totalPoints = 0;

        try (BufferedReader reader = new BufferedReader(new
FileReader(filePath))) {
            String line;

            while ((line = reader.readLine()) != null) {
                for (String term : phishingTerms) {
                    if (line.contains(term)) {
                        int points = getPoints(term);
                        totalPoints += points;
                        System.out.println("Se encontró el término: " +
term + " (" + points + " puntos)");
                    }
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println("Puntuación total de phishing: " +
totalPoints);
    }

    private static int getPoints(String term) {
        // Asignar valor de puntos a cada término o frase
        switch (term) {
            case "Urgente":
            case "Verificación de cuenta":
            case "Actualice su información":
            case "Premio en efectivo":
            case "Gane un viaje gratis":
            case "Confirme su identidad":
            case "Problema con su cuenta":
            case "Gane dinero desde casa":
            case "Phishing":
            case "Gane un millón de dólares":
                return 2;
            case "Gane dinero rápido":
            case "Pago pendiente":
            case "Actualización de software":
            case "Problema con su pedido":
            case "Alerta de seguridad":
            case "Verificación de seguridad":
            case "Restricción de cuenta":
            case "Restablecer contraseña":
                return 3;
            default:
                return 1;
        }
    }
}
```

4. Para cada ocurrencia de una palabra clave o frase dentro del archivo de texto, agregue el valor del punto asignado para esa palabra o frase al total de puntos acumulados.

```
public class PhishingDetector {
    public static void main(String[] args) {
        String filePath = "ruta/al/archivo.txt"; // Ruta del archivo de
        texto a explorar

        // Palabras, frases y nombres de compañías a buscar
        String[] phishingTerms = {
            "Urgente", "Verificación de cuenta", "Actualice su
información", "Gane dinero rápido",
            "Premio en efectivo", "Problema de seguridad",
            "Contraseña", "Cuenta suspendida",
            "Tarjeta de crédito", "Gane un viaje gratis", "Confirme
su identidad", "Oferta limitada",
            "Inicie sesión para reclamar", "Pago pendiente",
            "Actualización de software",
            "Problema con su pedido", "Alerta de seguridad",
            "Verificación de seguridad",
            "Restricción de cuenta", "Banco", "Tarjeta bancaria",
            "Phishing", "Premio de lotería",
            "Gane un iPhone", "Actualizar sus datos", "Problema con
su cuenta", "Gane dinero desde casa",
            "Oferta exclusiva", "Restablecer contraseña", "Gane un
millón de dólares"
        };

        int totalPoints = 0;

        try (BufferedReader reader = new BufferedReader(new
        FileReader(filePath))) {
            String line;

            while ((line = reader.readLine()) != null) {
                for (String term : phishingTerms) {
                    if (line.contains(term)) {
                        int points = getPoints(term);
                        totalPoints += points;
                        System.out.println("Se encontró el término: " +
                        term + " (" + points + " puntos)");
                    }
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println("Puntuación total de phishing: " +
totalPoints);
    }

    private static int getPoints(String term) {
        // Asignar valor de puntos a cada término o frase
        switch (term) {
            case "Urgente":
            case "Verificación de cuenta":
            case "Actualice su información":
            case "Premio en efectivo":
            case "Gane un viaje gratis":
            case "Confirme su identidad":
            case "Problema con su cuenta":
            case "Gane dinero desde casa":
            case "Phishing":
            case "Gane un millón de dólares":
                return 2;
            case "Gane dinero rápido":
            case "Pago pendiente":
            case "Actualización de software":
            case "Problema con su pedido":
            case "Alerta de seguridad":
            case "Verificación de seguridad":
            case "Restricción de cuenta":
            case "Restablecer contraseña":
                return 3;
            default:
                return 1;
        }
    }
}
```

5.Además, para cada palabra clave o frase encontrada, deberá imprimir en pantalla una línea con la palabra o frase, el número de ocurrencias y el total de puntos

```
public class PhishingDetector {
    public static void main(String[] args) {
        String filePath = "ruta/al/archivo.txt"; // Ruta del archivo de
        texto a explorar

        // Palabras, frases y nombres de compañías a buscar
        String[] phishingTerms = {
            "Urgente", "Verificación de cuenta", "Actualice su
información", "Gane dinero rápido",
            "Premio en efectivo", "Problema de seguridad",
            "Contraseña", "Cuenta suspendida",
            "Tarjeta de crédito", "Gane un viaje gratis", "Confirme
su identidad", "Oferta limitada",
            "Inicie sesión para reclamar", "Pago pendiente",
            "Actualización de software",
            "Problema con su pedido", "Alerta de seguridad",
            "Verificación de seguridad",
            "Restricción de cuenta", "Banco", "Tarjeta bancaria",
            "Phishing", "Premio de lotería",
            "Gane un iPhone", "Actualizar sus datos", "Problema con
su cuenta", "Gane dinero desde casa",
            "Oferta exclusiva", "Restablecer contraseña", "Gane un
millón de dólares"
        };

        Map<String, Integer> termOccurrences = new HashMap<>();
        int totalPoints = 0;

        try (BufferedReader reader = new BufferedReader(new
FileReader(filePath))) {
            String line;

            while ((line = reader.readLine()) != null) {
                for (String term : phishingTerms) {
                    if (line.contains(term)) {
                        int points = getPoints(term);
                        totalPoints += points;

                        if (termOccurrences.containsKey(term)) {
                            termOccurrences.put(term,
termOccurrences.get(term) + 1);
                        } else {
                            termOccurrences.put(term, 1);
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
}
} catch (IOException e) {
    e.printStackTrace();
}

for (Map.Entry<String, Integer> entry :
termOccurrences.entrySet()) {
    String term = entry.getKey();
    int occurrences = entry.getValue();
    int points = getPoints(term);

    System.out.println("Término: " + term + " - Ocurrencias: " +
occurrences + " - Puntos: " + points);
}

System.out.println("Puntuación total de phishing: " +
totalPoints);
}

private static int getPoints(String term) {
    // Asignar valor de puntos a cada término o frase
    switch (term) {
        case "Urgente":
        case "Verificación de cuenta":
        case "Actualice su información":
        case "Premio en efectivo":
        case "Gane un viaje gratis":
        case "Confirme su identidad":
        case "Problema con su cuenta":
        case "Gane dinero desde casa":
        case "Phishing":
        case "Gane un millón de dólares":
            return 2;
        case "Gane dinero rápido":
        case "Pago pendiente":
        case "Actualización de software":
        case "Problema con su pedido":
        case "Alerta de seguridad":
        case "Verificación de seguridad":
        case "Restricción de cuenta":
        case "Restablecer contraseña":
            return 3;
        default:
            return 1;
    }
}
}
}

```



6.Finalmente muestre el total de puntos para todo el mensaje.

```
public class PhishingDetector {
    public static void main(String[] args) {
        String filePath = "ruta/al/archivo.txt"; // Ruta del archivo de
        texto a explorar

        // Palabras, frases y nombres de compañías a buscar
        String[] phishingTerms = {
            "Urgente", "Verificación de cuenta", "Actualice su
información", "Gane dinero rápido",
            "Premio en efectivo", "Problema de seguridad",
            "Contraseña", "Cuenta suspendida",
            "Tarjeta de crédito", "Gane un viaje gratis", "Confirme
su identidad", "Oferta limitada",
            "Inicie sesión para reclamar", "Pago pendiente",
            "Actualización de software",
            "Problema con su pedido", "Alerta de seguridad",
            "Verificación de seguridad",
            "Restricción de cuenta", "Banco", "Tarjeta bancaria",
            "Phishing", "Premio de lotería",
            "Gane un iPhone", "Actualizar sus datos", "Problema con
su cuenta", "Gane dinero desde casa",
            "Oferta exclusiva", "Restablecer contraseña", "Gane un
millón de dólares"
        };

        Map<String, Integer> termOccurrences = new HashMap<>();
        int totalPoints = 0;

        try (BufferedReader reader = new BufferedReader(new
FileReader(filePath))) {
            String line;

            while ((line = reader.readLine()) != null) {
                for (String term : phishingTerms) {
                    if (line.contains(term)) {
                        int points = getPoints(term);
                        totalPoints += points;

                        if (termOccurrences.containsKey(term)) {
                            termOccurrences.put(term,
termOccurrences.get(term) + 1);
                        } else {
                            termOccurrences.put(term, 1);
                        }
                    }
                }
            }
        }
    }
}
```

```

    } catch (IOException e) {
        e.printStackTrace();
    }

    for (Map.Entry<String, Integer> entry :
termOccurrences.entrySet()) {
        String term = entry.getKey();
        int occurrences = entry.getValue();
        int points = getPoints(term);

        System.out.println("Término: " + term + " - Ocurrencias: " +
occurrences + " - Puntos: " + (points * occurrences));
    }

    System.out.println("Total de puntos para todo el mensaje: " +
totalPoints);
}

private static int getPoints(String term) {
    // Asignar valor de puntos a cada término o frase
    switch (term) {
        case "Urgente":
        case "Verificación de cuenta":
        case "Actualice su información":
        case "Premio en efectivo":
        case "Gane un viaje gratis":
        case "Confirme su identidad":
        case "Problema con su cuenta":
        case "Gane dinero desde casa":
        case "Phishing":
        case "Gane un millón de dólares":
            return 2;
        case "Gane dinero rápido":
        case "Pago pendiente":
        case "Actualización de software":
        case "Problema con su pedido":
        case "Alerta de seguridad":
        case "Verificación de seguridad":
        case "Restricción de cuenta":
        case "Restablecer contraseña":
            return 3;
        default:
            return 1;
    }
}
}

```