IDGS

Ingenieria en Desarrollo y Gestión de Software

Portafolio de Evidencias

Seguridad en el Desarrollo de Aplicaciones

Alumno: Jesús Alonso Cornejo Galindo

Profesor: Héctor Hugo Domínguez Jaime

Grupo: IDGS-81

Índice

Unidad 1 3
Practica_#1_Tipos de ataques 3
Rubrica 35
Examen 36
Resumen #2_Estándares de Codificación 37
Rubrica 63
Unidad 2 64
Resumen #1 Frameworks del Frontend y del Backend 64
Rubrica 81

Unidad 1

Practica_#1_Tipos de ataques

Instrumento	***Resumen***

Alumno: Jesús Alonso Cornejo Galindo.	**Fecha: 08/02/25**
Carrera: Ingeniería en Desarrollo y Gestión de Software	**Grupo: IDGS81**
Asignatura: Seguridad en el Desarrollo de Aplicaciones	**Unidad temática: Principios de codificación segura 1**
Profesor: MCE, Héctor Hugo Domínguez Jaime	

I. Ejercicios a resolver:

El alumno implementará un formato de Práctica de Ejercicios, donde exponga ejemplos reales (especificando lugar, fecha, noticia, imágenes o video y URL de donde obtuvo la información en formato APA) de los diversos tipos de ataques vistos en esta presentación, así mismo, agregará un apartado en cada uno de ellos, donde explicará detalladamente la forma de protección y prevención que tuvieron que tener para prevenir los ataques.

Titulo

"Ejemplos Reales de Ataques y Medidas de Prevención"

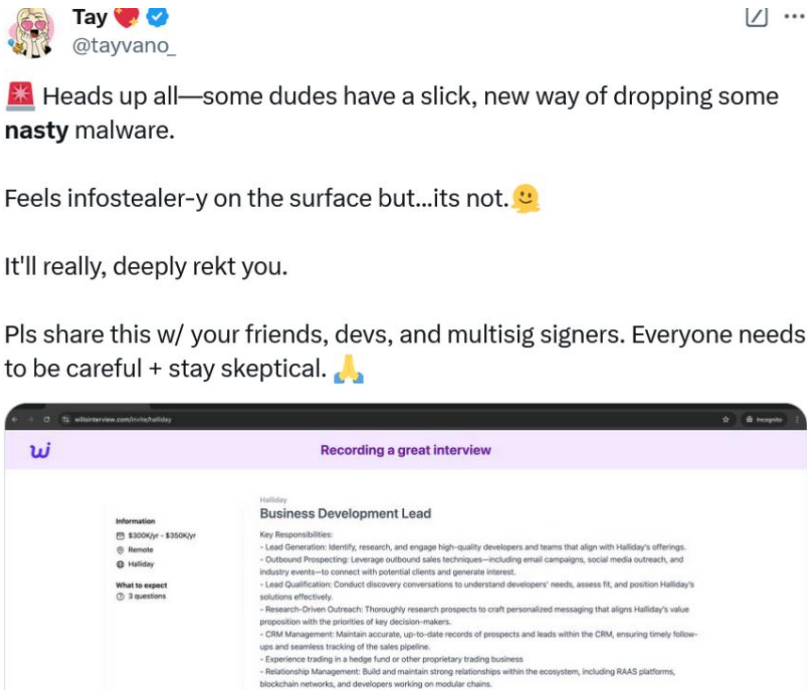
Introducción

En la actualidad, la ciberseguridad es un asunto que nos involucra a todos. A medida que la tecnología avanza y la cantidad de información que manejamos en línea crece, también lo hacen los ataques cibernéticos. Estos no son solo problemas que afectan a grandes empresas o gobiernos; cualquier persona puede convertirse en una víctima. Los ciberdelincuentes tienen diversas maneras de operar, desde virus y troyanos hasta fraudes diseñados para robar datos personales o incluso dinero. Ataques como el troyano Zeus, el spyware Pegasus o el ransomware RobbinHood han causado estragos a nivel mundial, impactando a miles de personas y organizaciones, y poniendo en riesgo la privacidad y seguridad de la información más sensible. Además, con el auge del teletrabajo y el uso generalizado de dispositivos móviles, las amenazas se han diversificado y ampliado, lo que hace aún más difícil anticipar y prevenir estos ataques. Los ciberdelincuentes utilizan tácticas cada vez más sofisticadas, como el phishing y el whaling, que explotan las debilidades humanas, como la confianza o el descuido. Por ello, es fundamental entender cómo funcionan estos ataques, qué herramientas emplean los criminales y qué medidas se pueden tomar para prevenirlos. La seguridad en internet no es solo responsabilidad de los expertos en tecnología; todos podemos implementar acciones para proteger nuestra información personal, como usar contraseñas seguras, estar alertas ante correos electrónicos sospechosos y mantener nuestros dispositivos actualizados. Solo a través de una conciencia colectiva y la adopción de buenas prácticas podemos enfrentar los riesgos de un mundo digital cada vez más vulnerable.

Malware (Software Malicioso)

Según Lindrea (2024), los ciberdelincuentes que operan en el mundo de las criptomonedas han ideado una táctica innovadora para engañar a sus víctimas y propagar malware en sus dispositivos. Estos hackers se hacen pasar por reclutadores de empresas de criptomonedas de prestigio, ofreciendo atractivas ofertas laborales con salarios de entre 200,000 y 350,000 dólares, lo que despierta el interés de sus víctimas. Sin embargo, la verdadera intención de estos estafadores es inducir a las personas a descargar software malicioso, lo que les permite tomar control remoto de sus dispositivos. Esta acción les otorga acceso a las billeteras digitales de las víctimas, lo que puede resultar en el robo de criptomonedas, así como otros daños importantes. Según Taylor Monahan, conocida en X como Tay, esta modalidad de fraude representa una evolución significativa en las técnicas de ingeniería social utilizadas por los delincuentes, que buscan explotar la confianza y la codicia de sus objetivos.

4o mini



En lugar de intentar que la víctima abra un archivo PDF con malware o descargue un software de videollamadas disfrazado, este nuevo enfoque se centra en hacer que la persona siga instrucciones para solucionar un supuesto problema con el acceso al micrófono y la cámara. Como explicó Monahan, “Si sigues sus instrucciones, estás perdido”.

El proceso comienza con una serie de preguntas en la entrevista que requieren respuestas detalladas, seguido de una última pregunta que debe ser grabada en video a través de "Willo | Video Interviewing". Durante este proceso, la víctima experimenta un problema al intentar habilitar el acceso al micrófono y la cámara, siendo informada de un error de caché y luego recibiendo una "solución" para resolverlo. Monahan señala que, tras seguir estas instrucciones, Chrome pedirá a la víctima que actualice o reinicie el navegador para "corregir el problema". Sin embargo, en realidad, el problema no se resuelve, y la víctima queda completamente comprometida.

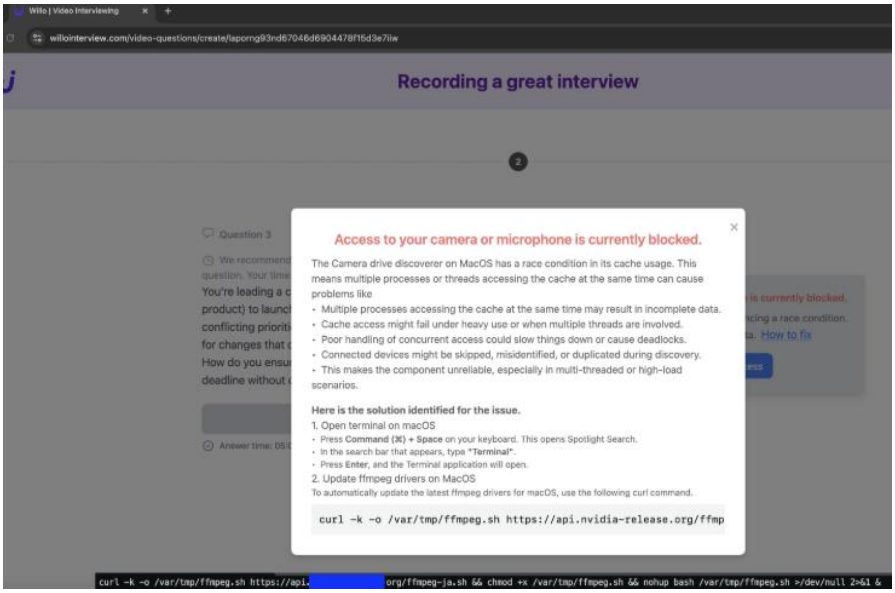


Imagen del mensaje que reciben las víctimas al intentar otorgar acceso a la cámara y al micrófono. Fuente: Taylor Monahan.

Monahan explicó que el malware otorga a los atacantes acceso "a través de una puerta trasera" a los dispositivos de las víctimas, lo que les permite robar criptomonedas. "Al final, te destruirán por

cualquier medio necesario", agregó, subrayando que estos ataques son efectivos en sistemas operativos como Mac, Windows y Linux. El troyano Zeus es

Además, Monahan señaló que los falsos reclutadores están contactando a las víctimas de manera sorpresiva a través de LinkedIn, ofreciendo desde puestos de gerente de desarrollo empresarial hasta analistas e investigadores en empresas de criptomonedas reconocidas, como Gemini y Kraken. Los hackers también se están acercando a las personas a través de plataformas de freelancers, Discord y Telegram.

Las preguntas que se les hacen en las entrevistas incluyen temas como las tendencias de criptomonedas que la víctima considera más relevantes para la industria en el próximo año, así como cómo un representante de desarrollo empresarial podría expandir las asociaciones de un exchange de criptomonedas en regiones como el sudeste asiático o América Latina, todo dentro de un “presupuesto limitado”.

Protección y Prevención:

Verificar la legitimidad de la oferta

- Investigue a fondo la empresa que ofrece el empleo.
- Comuníquese directamente con la empresa a través de canales oficiales para confirmar la autenticidad de la oferta.

Ser cauteloso con enlaces y archivos adjuntos

- Evite descargar software o abrir archivos adjuntos de fuentes no verificadas.
- No siga instrucciones que requieran cambiar configuraciones de seguridad o instalar programas desconocidos.

Mantener la seguridad del sistema

- Actualice regularmente su software antivirus y antimalware.

- Asegúrese de que su sistema operativo y aplicaciones estén al día con los últimos parches de seguridad.

Educarse sobre tácticas de ingeniería social

- Aprenda a identificar señales de estafas, como ofertas de trabajo no solicitadas con salarios inusualmente altos.
- Desconfíe de solicitudes inusuales durante el proceso de entrevista, especialmente si implican instalar software o proporcionar información sensible.

Troyano

El troyano Zeus es un malware diseñado principalmente para sistemas operativos Microsoft Windows, y es comúnmente utilizado para robar información financiera. Su funcionamiento se basa en un software que sigue y graba las pulsaciones de teclas del usuario, permitiéndole obtener credenciales bancarias de manera furtiva (Zeus: El malware troyano, 2017, diciembre 7).

El troyano Zeus suele infectar a sus víctimas a través de correos electrónicos de phishing o anuncios en línea maliciosos.

El troyano Zeus fue identificado por primera vez en 2007, cuando los ciberdelincuentes lo emplearon para robar información del Departamento de Transporte de los Estados Unidos. Sin embargo, fue en 2009 cuando se descubrió que Zeus había comprometido más de 74.000 cuentas FTP de importantes empresas como Bank of America, NASA, Monster.com, ABC, Oracle, Play.com, Cisco, Amazon y BusinessWeek, entre otras. Este ataque se considera uno de los más conocidos en el mundo del malware, afectando a millones de dispositivos y dando lugar al desarrollo de nuevas variantes. Algunas

de estas versiones han logrado evadir la seguridad de autenticación de dos factores. Aunque en sus inicios solo afectaba a sistemas operativos Windows, las versiones más recientes también impactan dispositivos móviles con sistemas como Symbian, BlackBerry y Android.

Protección y Prevención:

Uso de soluciones antivirus y antimalware: Implementar software de seguridad confiable que detecte y elimine troyanos y otros programas maliciosos.

Autenticación robusta: Activar autenticación multifactor para proteger cuentas bancarias y servicios en línea.

Evitar enlaces y archivos sospechosos: No descargar archivos adjuntos ni hacer clic en enlaces de correos electrónicos no solicitados.

Segmentación de redes: Separar los sistemas críticos para limitar el impacto de posibles infecciones.

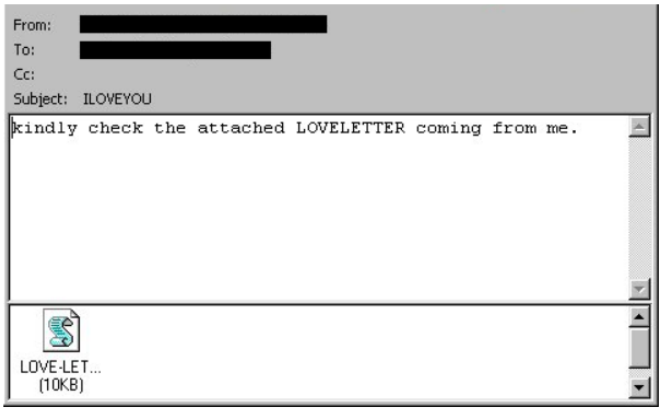
Educación en ciberseguridad: Capacitar a los usuarios sobre la identificación de intentos de phishing y prácticas seguras en línea.

Monitorización de actividad: Implementar herramientas para detectar comportamientos inusuales en el sistema y bloquear accesos no autorizados.

Virus:

ILOVEYOU

García (2018) relata que en los años 2000, un virus informático llamado I LOVE YOU surgió en Filipinas y se propagó rápidamente a través de correos electrónicos en Asia, Europa y América, en tan solo cinco horas. El mensaje, con el asunto "ILOVEYOU", contenía un archivo adjunto que parecía una carta de amor, pero en realidad era un script malicioso. Al abrirlo, el virus infectaba los contactos de Outlook de la víctima, hacía inaccesibles los archivos y robaba las contraseñas. Este ataque afectó a millones de computadoras en todo el mundo, resaltando la relevancia de la ciberseguridad.



Un correo con el gusano ILOVEYOU se parecía a esto en el antiguo correo electrónico de Microsoft. Fuente

Cualquiera que sea la persona, el archivo adjunto llamado "CARTA-DE-AMOR-PARA-TI.TXT.VBS" ciertamente atrae tu atención, por lo que decides hacer clic en él. Al principio, parece que no sucede nada. Sin embargo, más tarde descubres que documentos importantes en tu disco duro se han dañado irreversiblemente, y que se han enviado múltiples cartas de amor, similares a la que abriste, a todos tus contactos.

El virus afectaba archivos con extensiones .VBS y .VBE, sobrescribiéndolos con su código. Además, eliminaba archivos con extensiones como .JS, .JSE, .CSS, .WSH, .SCT y .HTA, creando nuevos archivos con el mismo nombre pero con la extensión .VBS. También localizaba archivos de imagen y audio (.JPG, .JPEG, .MP3 y .MP2), los eliminaba y los reemplazaba con archivos maliciosos que llevaban el mismo nombre pero con la extensión .VBS. En otras palabras, destruía imágenes y sonidos, pero convertía esos mismos nombres de archivos en vectores de malware.

Se cree que Onel de Guzmán fue el creador de este virus, quien admitió haberlo propagado "por accidente". Según se informó, I LOVE YOU formaba parte de su tesis universitaria, que abordaba cómo robar códigos secretos por Internet o cómo acceder y controlar computadoras ajenas.

Protección y Prevención:

Mantener el software actualizado: Asegúrese de que su sistema operativo y todas las aplicaciones estén al día con las últimas actualizaciones de seguridad.

Utilizar programas antivirus confiables: Instale y mantenga actualizado un software antivirus para detectar y eliminar posibles amenazas.

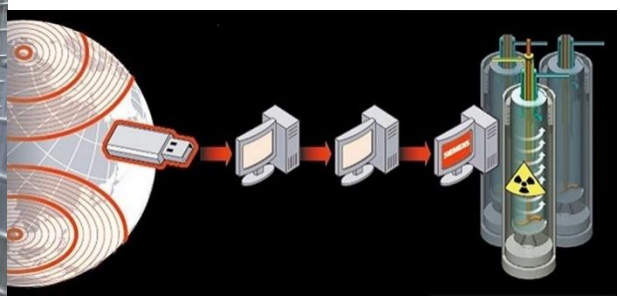
Ser cauteloso con correos electrónicos y archivos adjuntos: No abra correos electrónicos ni descargue archivos de remitentes desconocidos o sospechosos.

Evitar el uso de redes Wi-Fi públicas: Las redes públicas pueden ser vulnerables; prefiera conexiones seguras y privadas.

Realizar copias de seguridad periódicas: Haga respaldos regulares de sus datos importantes para prevenir pérdidas en caso de infección.

Gusanos:

Menendez, R. (2020, June 26). *El análisis definitivo de Stuxnet (En español) probablemente el virus más peligroso de la historia*. LinkedIn. Stuxnet es considerado uno de los virus más sofisticados y peligrosos de la historia, diseñado específicamente para atacar sistemas industriales, particularmente aquellos utilizados en la planta de enriquecimiento de uranio en Irán. Su capacidad para manipular equipos físicos y su impacto en las operaciones industriales lo han convertido en un hito en la ciberguerra.



Para comprender el ataque de Stuxnet, no basta con analizar únicamente el código o determinar si se aprovechó una vulnerabilidad de tipo Zero-Day, sino que es esencial entender el contexto físico, el objetivo principal del ataque, el diseño de la planta industrial y los parámetros de proceso involucrados. Según Schneider, en un ataque denominado Cyber-Físico, existen tres capas:

1. Capa de TI, que se utiliza para propagar el malware, conocida como *Propagación*.
2. Capa de control industrial de sistemas, que manipula (sin interrumpir) el control de las máquinas industriales, denominada *Manipulación*.
3. Capa física, donde ocurre el daño real, afectando sistemas eléctricos, válvulas de control, etc., y se denomina *Daño o Explotación*.

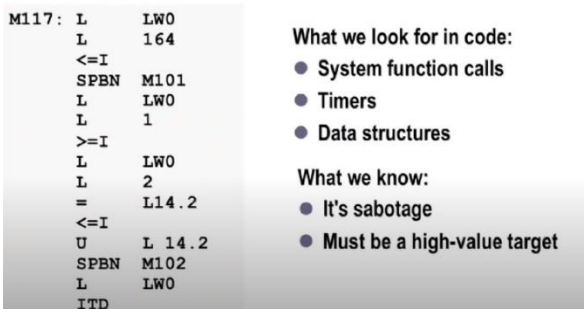
En 2007, alguien subió a la plataforma Virtustotal.com un código que resultaría ser la primera variante de Stuxnet. Este código contenía un *payload* dirigido a los sistemas de protección en cascada (CPS - Cascade Protection System). Según el informe, la centrifugadora IR-1 en Irán, aunque tenía fallas en su diseño, era crucial para el proceso de extracción de uranio.

Los sistemas de protección no operan durante el funcionamiento normal, sino que están diseñados para detectar anomalías y prevenir daños a equipos o peligros para los operadores o el medio ambiente. Irán tenía un sistema de protección en cascada para sus centrifugadoras. Este sistema consistía en dos capas: la capa inferior, ubicada a nivel de la centrifugadora, tenía tres válvulas de cierre rápido, que aislarían la centrifugadora afectada sin detener el proceso. En la pantalla de monitoreo central, se usaban "puntos de colores" (verdes y grises) para indicar el estado de las centrifugadoras, pero si se generaban alarmas constantes, las válvulas se aislaban, lo que afectaba la operación normal de la planta.

4o mini

Figure 3: Former president Ahmadinejad looking at SCADA screens in the control room at Natanz in 2008. The screen facing the photographer shows that two centrifuges are isolated, indicating a defect, but that doesn't prevent the respective cascade from continuing operation (highlighting in red not in the original)

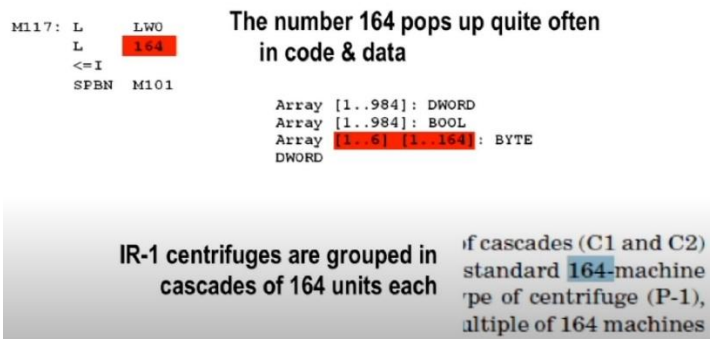
Ralph Langnet señala que este ataque fue completamente dirigido y meticulosamente planificado. Al analizar el código del malware, comenzaron a descubrir elementos realmente sorprendentes.



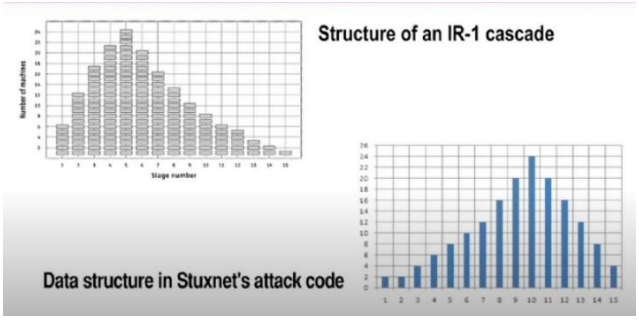
En el análisis de una muestra de código de aproximadamente 15,000 líneas, que se asemejaba al lenguaje ensamblador, se encontraron llamados a funciones, temporizadores y estructuras de datos cuidadosamente diseñadas. ¿Qué buscaban? Sabotaje deliberado. ¿De dónde provenía? Probablemente de Irán, ya que la mayoría de los reportes fueron originados en ese país. El equipo de Langnet convocó a expertos en centrifugadoras y plantas de energía, quienes, tras analizar y estudiar el código, concluyeron que el problema principal fue causado por el rotor, una parte móvil dentro de la centrifugadora.

Figure 9: President Ahmadinejad holding a carbon fiber centrifuge rotor during his 2008 press tour at Natanz. This rotor is for the next-generation IR-2 centrifuge. Rotors used in the IR-1 that was attacked by Stuxnet are taller and built from metal.

Se determinó que, al alterar el rotor, técnicamente se podría "romper" el sistema e incluso causar la explosión de la centrifugadora mencionada anteriormente. Además, al analizar el código del malware, se encontró de manera curiosa el número 164, que, después de ser investigado, resultó ser la cantidad de centrifugadoras por cada cascada.

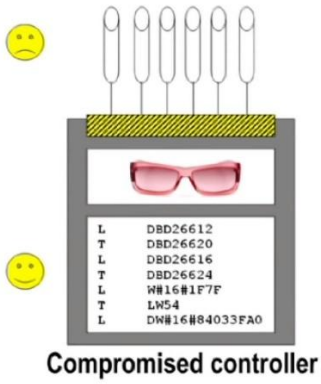


Además, descubrieron que cada centrifugadora está dividida en 15 partes, conocidas como etapas, y dentro de la estructura del código del malware encontraron algo sorprendentemente similar en la organización de los datos.



Para explicarlo de manera clara:

Lo que denominan "caja gris" en la imagen hace referencia a un sistema en el que el malware manipula las señales de los sensores, como los de presión o vibración, que están conectados a las válvulas de las centrifugadoras. El malware intercepta las órdenes de estos sensores y las modifica, enviando código legítimo para que no se detecten anomalías en el sistema. Esto significa que, aunque los datos de entrada son alterados, el sistema de control (SCADA) no sospecha que haya algo fuera de lugar, ya que recibe información que parece completamente normal. De esta manera, el ataque se vuelve difícil de detectar y permite que el sabotaje ocurra sin ser identificado.



Compromised controller

La imagen que muestra el tráfico de datos entre un sistema SCADA infectado por Stuxnet y un controlador ilustra cómo este tráfico se produce periódicamente, cada cinco segundos. Lo crucial de este comportamiento es que, al ser tan regular y estructurado, se volvía prácticamente indetectable para los expertos en seguridad de Sistemas de Control Industrial (ICS). Este tráfico aparentemente normal no levantaba sospechas, lo que permitía al malware operar de manera encubierta sin ser identificado, dificultando aún más la detección del ataque.

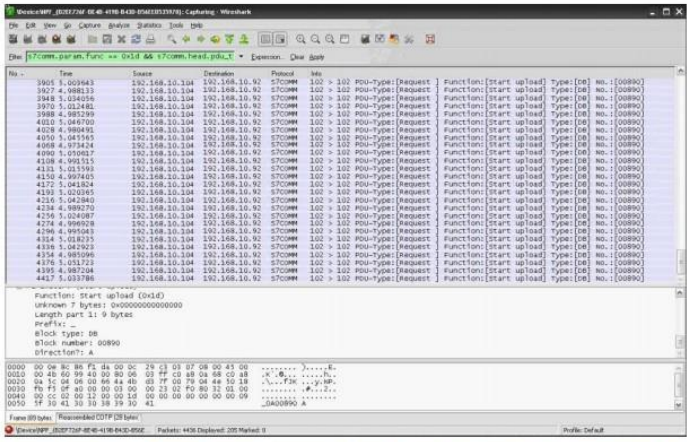


Figure 11: Data traffic between a Stuxnet-infected WinCC SCADA system and a controller, occurring periodically every five seconds, as captured in a properly equipped forensic lab. This traffic simply could not be missed or misinterpreted by ICS security experts; it points to a cyber attack at the controller's application layer

Aunque ningún país ha confirmado públicamente su participación en el desarrollo de Stuxnet, se cree que fue creado conjuntamente por Estados Unidos e Israel como un arma cibernética. Diversos informes indican que Stuxnet causó daños a cerca del 20% de las centrifugadoras nucleares de Irán,

además de infectar más de 200,000 computadoras y hacer que al menos 1,000 terminales dejaran de funcionar.

Protección y Prevención:

Actualización de sistemas: Mantener el software y los sistemas operativos actualizados para corregir vulnerabilidades conocidas.

Segmentación de redes: Dividir la red en segmentos aislados para limitar la propagación de posibles infecciones.

Control de dispositivos externos: Restringir el uso de dispositivos USB y otros medios extraíbles para prevenir la introducción de malware.

Monitoreo continuo: Implementar sistemas de detección de intrusiones para identificar actividades sospechosas en tiempo real.

Capacitación del personal: Educar a los empleados sobre prácticas de seguridad y concienciarlos sobre posibles amenazas.

Spyware:

Según Jiménez (2022), los teléfonos del presidente del gobierno de España, Pedro Sánchez, y de la ministra de Defensa, Margarita Robles, fueron infectados con el spyware Pegasus en un ataque externo confirmado en 2021.

Pegasus es una aplicación espía altamente sofisticada, utilizada comúnmente para vigilar a figuras políticas y sociales de alto perfil. Esta herramienta es sumamente eficaz y generalmente se comercializa para ser utilizada por gobiernos, según afirman sus desarrolladores. En una rueda de prensa urgente, Félix Bolaños, ministro de Presidencia, confirmó que los teléfonos del presidente Pedro Sánchez y de la ministra de Defensa, Margarita Robles, habían sido infectados con Pegasus, basándose en dos informes del Centro Criptológico Nacional que señalaron que se trataba de un ataque externo.

El espionaje ha alcanzado niveles muy altos. Después del escándalo relacionado con el uso de Pegasus para espiar a 65 políticos independentistas catalanes, el Centro Criptológico Nacional (CCN) investigó posibles ataques en los teléfonos del Gobierno. Según el Ministro de Presidencia, el primer informe del CCN-CERT confirma que en mayo de 2021, dos intrusiones afectaron al teléfono de Pedro Sánchez, y en junio de ese mismo año, otro ataque tuvo lugar en el teléfono de la Ministra de Defensa, Margarita

Robles. Gracias a Pegasus, los atacantes lograron extraer 2,6 gigabytes del teléfono de Sánchez en la primera intrusión y 130 megabytes en la segunda. En el caso de Robles, únicamente se extrajeron 9 megabytes. Aunque no se ha revelado el contenido exacto de estos datos, esta mañana la Abogacía del Estado presentó una denuncia ante el juez de guardia de la Audiencia Nacional, solicitando una investigación sobre este espionaje, calificado por el Gobierno como "ilícito y externo".

Pegasus es un software utilizado para espiar a personalidades de alto nivel, y según la versión oficial de la empresa NSO, el producto está diseñado para ayudar a los gobiernos a prevenir actos terroristas, desarticular redes delictivas y localizar personas desaparecidas. Sin embargo, la falta de transparencia sobre las compras de Pegasus ha generado controversia, ya que se sabe que alrededor de 1.400 personas han sido espiadas con esta herramienta, pero se desconoce quiénes han adquirido el software.

Protección y Prevención:

Protección técnica:

- Mantener dispositivos y aplicaciones actualizados.
- Usar cifrado en comunicaciones y VPN seguras.
- Implementar sistemas de detección y monitoreo de intrusiones.
- Restringir permisos y acceso a datos sensibles.
- Realizar auditorías de seguridad y análisis forense preventivo.

Medidas organizativas:

- Establecer protocolos de uso para dispositivos oficiales.
- Capacitar al personal en ciberseguridad.
- Usar contraseñas fuertes y autenticación multifactorial.
- Tener un plan de respuesta rápida ante incidentes.

- Evaluar la seguridad de tecnologías y proveedores.

Recomendaciones críticas:

- Utilizar dispositivos aislados para información confidencial.
- Supervisar proveedores y contratistas.
- Cooperar a nivel internacional para prevenir espionaje.

AdWare:

Según Ruiz (2019), SimBad es un malware para Android que estuvo presente en más de 200 aplicaciones disponibles en Google Play.

El equipo de seguridad móvil de CheckPoint descubrió una nueva campaña de adware en Google Play Store, llamada SimBad. Este malware fue encontrado en más de 200 aplicaciones, las cuales acumularon cerca de 150 millones de descargas en total.

El código malicioso se introdujo a través del SDK RXDrioder, proporcionado por addroider[.]com, un kit de desarrollo relacionado con anuncios. Este SDK permitió que diversas aplicaciones, desarrolladas por diferentes programadores, incluyeran el malware sin su conocimiento.

Una vez que el usuario instala una de estas aplicaciones, SimBad se conecta a un servidor de Comando y Control (C&C) para recibir instrucciones. Entre sus funciones, se incluyen ocultar el ícono de la aplicación (lo que dificulta su desinstalación), mostrar anuncios en segundo plano, abrir un navegador con URLs predeterminadas para realizar ataques de phishing, redirigir a aplicaciones de markets como Google Play o 9Apps para mostrar resultados de búsqueda específicos, o incluso descargar nuevas aplicaciones desde un servidor remoto.

Tras recibir el aviso, Google eliminó todas las aplicaciones afectadas de su tienda, aunque el malware podría seguir disponible en otros mercados. Se recomienda revisar la lista de aplicaciones comprometidas, muchas de las cuales son juegos de simulación, que está disponible en el informe de CheckPoint.

Protección y Prevención:

Desarrolladores:

- Evaluar la reputación y funcionalidad de los SDKs antes de integrarlos.
- Usar herramientas para analizar el código de la app y sus dependencias.

Google Play Store:

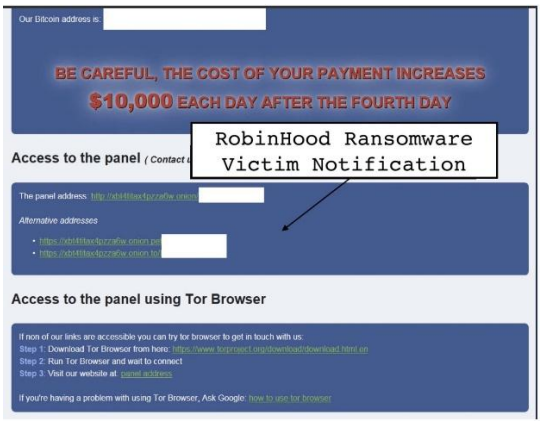
- Mejorar la revisión de apps con herramientas de detección de malware.
- Realizar auditorías periódicas de las aplicaciones publicadas.

Usuarios:

- Revisar permisos y reseñas de las aplicaciones antes de instalarlas.
- Mantener los dispositivos y apps actualizados para recibir parches de seguridad.

Ransomware:

Según RobbinHood (2019), "este malware fue detectado por primera vez en el año 2019 después de explotar varios sistemas en Maryland y Greenville. Las autoridades denunciaron extorsiones acumulativas de casi 1,5 millones de dólares. Sin embargo, los estafadores no usaron el dinero para ayudar a los pobres o ninguna otra meta altruista como el famoso personaje del folklore inglés. En lugar de eso, el ransomware RobbinHood fue utilizado para extorsionar a organizaciones y exigir pagos en criptomonedas a cambio de no divulgar o destruir datos críticos".



El ransomware RobbinHood (identificado por Trend Micro como Ransom.Win32.ROBBINHOOD.A), que se propaga principalmente a través de servicios de escritorio remoto comprometidos o de otros malwares, se ha ganado una mala reputación al atacar a organizaciones y computadoras dentro de sus redes. Este malware utiliza su imagen temible para intimidar a las víctimas y presionarlas a pagar un rescate.

Según BleepingComputer, una nueva variante de RobbinHood emplea tácticas de miedo en su mensaje de rescate, alentando a las víctimas a investigar en línea sobre casos anteriores de personas que no pagaron y terminaron enfrentando mayores costos. La nota de rescate, hallada por Joakim Kennedy, advierte a las víctimas que los cibercriminales que controlan RobbinHood han logrado obtener acceso completo a sus sistemas y eludido las medidas de seguridad. Sin embargo, la forma exacta en que el malware entra en la red de las víctimas aún no se ha determinado.

Las víctimas de RobbinHood son amenazadas con un aumento del rescate en 10.000 dólares por día si no pagan dentro de los primeros cuatro días, y si no se abona el monto total en diez días, los archivos quedarán bloqueados de forma permanente tras eliminarse las claves y el panel de control.

Actualmente, no existe ninguna herramienta pública para descifrar los archivos afectados, y los atacantes se jactan de que la única forma de recuperar los archivos es pagar por el software de descifrado.

Además, los cibercriminales hacen referencia a víctimas previas de alto perfil, como las ciudades de Greenville y Baltimore, para mostrar cuán grave es la amenaza. Estas ciudades sufrieron grandes retrasos operativos debido al ataque de ransomware, con Baltimore City perdiendo unos 18,2 millones de dólares. Según el informe de seguridad de Trend Micro de la primera mitad de 2019, los ataques de ransomware aumentaron un 77 % en comparación con la segunda mitad de 2018, y la mayoría de las víctimas pertenecen a multinacionales, empresas y organizaciones gubernamentales.

Protección y Prevención:

Realizar copias de seguridad de los archivos periódicamente y garantizar la integridad de dichas copias de seguridad.

Mantener el sistema, la red, los servidores y los programas/aplicaciones actualizados y parcheados (o utilizar parches virtuales para sistemas o software heredados e integrados).

Aplicar el principio del mínimo privilegio para reducir la superficie de ataque, por ejemplo, asegurando el uso de herramientas de administración del sistema, restringiendo y deshabilitando componentes innecesarios u obsoletos y asignando solo los privilegios necesarios a las cuentas de usuario.

Doxing:

"Según Destape (2022), este joven argelino de 24 años fue buscado durante varios años por el FBI, acusado de robar más de 200 millones de dólares vulnerando los sistemas de seguridad online de 217 bancos."

Según Destape (2022), Hamza Bendelladj, un joven argelino de 24 años, fue perseguido durante varios años por el FBI debido a que se le acusaba de haber robado más de 200 millones de dólares al vulnerar los sistemas de seguridad en línea de 217 bancos. Además de cometer fraudes bancarios, Bendelladj también comercializó diversas versiones del virus SpyEye con otros desarrolladores, lo que permitió a otros ciberdelincuentes realizar delitos similares sobre otras víctimas. Según información proporcionada por el FBI, él ganaba aproximadamente 10 millones de dólares por cada operación delictiva, mientras recorría el mundo viajando en primera clase y disfrutando de los lujos exclusivos de los millonarios. La caída de Bendelladj ocurrió en 2013, cuando fue detenido en el aeropuerto de Tailandia y posteriormente extraditado a Estados Unidos, donde fue condenado a 15 años de cárcel y 3 años de libertad condicional.

Protección y Prevención:

Actualizar regularmente las políticas de privacidad y seguridad de los datos: Las empresas bancarias deben revisar de forma rutinaria sus políticas relacionadas a la privacidad y seguridad de la información.

Capacitar al personal en seguridad de datos: Es fundamental que todos los empleados que gestionen datos sensibles, así como el personal de TI, participen en una formación regular y obligatoria sobre ciberseguridad y protección de datos.

Encriptar: siempre la información sensible, como por ejemplo la información de cada cliente o usuario del banco y las cuentas.

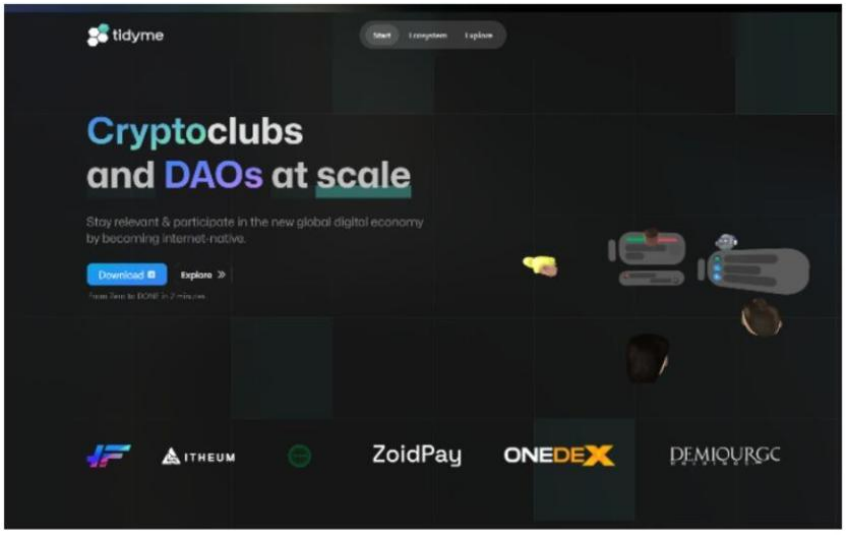
Phishing:

"Según De Frutos Sastre, De Frutos Sastre y De Frutos Sastre (2024), la campaña de phishing Tusk tiene como objetivo robar tanto información personal como criptomonedas de las víctimas."

La campaña de phishing Tusk, que busca robar tanto información personal como criptomonedas, pone de manifiesto el creciente peligro de las amenazas en línea. Es fundamental estar al tanto de la ciberseguridad, ya que los ciberdelincuentes están cada vez más activos, con el fin de vaciar cuentas bancarias y apoderarse de criptomonedas, como ocurre con esta nueva campaña.

El objetivo de los atacantes son personas a nivel mundial. Según expertos en ciberseguridad, este fraude está llevado a cabo por un grupo de ciberdelincuentes de habla rusa, quienes propagan malware diseñado para robar información y clipper malware (utilizado para robar criptomonedas).

El equipo Global de Respuesta a Emergencias (GERT) de Kaspersky ha detectado la campaña Tusk, dirigida a usuarios de Windows y macOS a nivel mundial. Esta campaña emplea sitios web fraudulentos que replican con precisión plataformas legítimas, con la intención de engañar a las víctimas y robar sus datos personales y criptomonedas. Recientemente, los atacantes han falsificado plataformas de criptomonedas, juegos en línea y traductores de inteligencia artificial.



Webs suplantadas
KASPERSKY

Este tipo de engaños suele presentar pequeñas variaciones en los elementos de las páginas falsas, como en el nombre o la URL, pero la campaña está muy bien diseñada y tiene una alta probabilidad de éxito. Si caes en la trampa y accedes a un sitio web que imita uno legítimo, podrías revelar información sensible, como las contraseñas privadas de tus monederos de criptomonedas, o incluso descargar malware.

Con esta información, los atacantes pueden acceder rápidamente a los monederos de criptomonedas de las víctimas y transferir sus fondos en segundos, utilizando herramientas comunes como infostealers para robar los datos.

“La correlación entre las diferentes partes de esta campaña y su infraestructura compartida sugiere una operación bien organizada, probablemente vinculada a un único actor o grupo con motivos financieros específicos. Además de las tres subcampañas dirigidas a criptomonedas, IA y juegos, nuestro Threat Intelligence Portal ha ayudado a identificar infraestructura para otros 16 temas, ya sean subcampañas anteriores, retiradas o nuevas que aún no se han lanzado. Esto demuestra la capacidad del actor de amenazas para adaptarse rápidamente a temas de tendencia y desplegar nuevas operaciones maliciosas. Subraya la necesidad crítica de soluciones de seguridad sólidas y mayor alfabetización en ciberseguridad para protegerse contra amenazas en constante evolución”, Según Ayman Shaaban, líder de la Unidad de Respuesta a Incidentes del GERT de Kaspersky.

Protección y Prevención:

Verificación de remitentes: Comprobar la autenticidad de los correos electrónicos y mensajes antes de interactuar con ellos.

No compartir información sensible: Evitar proporcionar datos personales o financieros a través de enlaces o formularios no verificados.

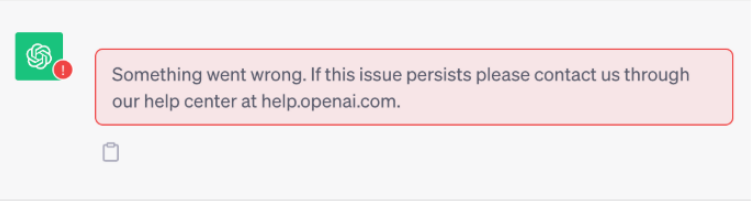
Educación: Capacitar a los usuarios para reconocer señales de phishing, como errores gramaticales o solicitudes urgentes de información.

Denegación de Servicio Distribuido (DDoS):

OpenAI confirma ataques DDoS que provocan interrupciones en el servicio ChatGPT

Según Gatlan (2023), OpenAI confirmó que los ataques DDoS fueron la causa de las interrupciones en el servicio de ChatGPT, reportadas desde Sudán en 2018.

Según OpenAI, los cortes periódicos en ChatGPT se deben a un patrón anómalo de tráfico relacionado con un ataque DDoS, y la empresa continúa trabajando para mitigar estos problemas. Los usuarios afectados experimentan mensajes de error como "algo parece haber salido mal" y "hubo un error al generar una respuesta". Este incidente sigue a una serie de interrupciones recientes, que incluyen una caída importante de ChatGPT y su API, además de problemas con tasas de error en Dall-E. OpenAI también indicó que la alta demanda está contribuyendo a estos problemas y solicitó a los usuarios paciencia mientras amplían sus sistemas.



Interrupción de ChatGPT (BleepingComputer)

Aunque OpenAI no ha confirmado oficialmente la autoría de los ataques DDoS, el grupo Anonymous Sudan se atribuyó la responsabilidad el miércoles, alegando que los ataques fueron motivados por el "sesgo general de la compañía hacia Israel y contra Palestina". Los atacantes anunciaron en su canal de Telegram que ChatGPT estaba inactivo en todo el mundo y pidieron que se reconociera que se trataba de un ataque DDoS. También confirmaron el uso de la botnet SkyNet, que ha estado operando desde octubre y recientemente incorporó soporte para ataques de capa de aplicación (DDoS de capa 7).

Los ataques de capa 7 son efectivos porque se enfocan en saturar los servicios con solicitudes masivas a nivel de aplicación, causando que los servicios se caigan al no poder procesarlas todas. Estos ataques ejercen presión sobre los recursos de red y servidores de los objetivos. A diferencia de los ataques de amplificación de DNS, que se enfocan en el consumo de ancho de banda, los ataques de capa 7 afectan directamente a los recursos del servidor.

En junio, Anonymous Sudan también llevó a cabo ataques DDoS de capa 7 contra servicios como Outlook.com, OneDrive y Azure Portal de Microsoft, lo que fue confirmado por la empresa. Microsoft rastrea estas actividades bajo el nombre de Storm-1359, destacando que el grupo utiliza tres tipos de ataques: inundaciones HTTP(S), omisión de caché y Slowloris.

Anonymous Sudan, que comenzó sus operaciones en enero de 2023, ha atacado a organizaciones y agencias gubernamentales globales. Sin embargo, algunos investigadores sugieren que podría tratarse de una falsa bandera y que el grupo podría tener vínculos con Rusia. OpenAI no ha respondido a las solicitudes de comentarios sobre estos ataques y las interrupciones actuales.

Protección y Prevención:

Infraestructura robusta: Implementar soluciones de balanceo de carga y servidores redundantes para manejar picos de tráfico.

Sistemas de detección: Utilizar herramientas que identifiquen y mitiguen tráfico malicioso en tiempo real.

Proveedores especializados: Contratar servicios de mitigación de DDoS que ofrezcan protección adicional.

Inyección SQL o SQL Injection:

Según Franceschi-Bicchierai y Franceschi-Bicchierai (2024), *"Bugs on Epic Games site allowed hackers to login to any 'Fortnite' player's account"*, los actores maliciosos aprovecharon vulnerabilidades en el sitio web de Epic Games para obtener acceso no autorizado a las cuentas de los

jugadores de Fortnite. Estos fallos permitieron que los hackers se conectaran a cualquier cuenta del juego sin autorización, comprometiendo la seguridad de los usuarios.

Un análisis de la empresa de ciberseguridad Check Point reveló que los ciberdelincuentes podían hackear cuentas de *Fortnite* aprovechando varios fallos en el sitio web de Epic Games. Los atacantes podían redirigir el tráfico de la página de inicio de sesión de Epic Games a otra página dentro del mismo sitio, lo que les permitía robar los tokens de inicio de sesión. Estos tokens, que funcionan como claves digitales, permiten a los jugadores acceder a sus cuentas en plataformas como Facebook, PlayStation Network, Xbox Live, Nintendo y Google+.

El ataque se iniciaba cuando un jugador de *Fortnite* hacía clic en un enlace malicioso. Al hacer esto, los hackers podían hacer que el servicio de inicio de sesión (como Facebook o Xbox Live) reenviara el token a un subdominio de Epic Games, que era vulnerable debido a dos fallos comunes en sitios web: inyección SQL y Cross-Site Scripting (XSS). Estos errores permitieron a los investigadores obtener el token de inicio de sesión de la víctima y acceder a la cuenta de *Fortnite*.

Epic Games confirmó la investigación, agradeciendo a Check Point por alertarlos sobre el problema. La compañía también recomendó a los jugadores utilizar contraseñas seguras y no compartir información de la cuenta. Además, los fallos fueron corregidos por Epic Games después de que se notificó el problema. Este incidente pone de relieve cómo los ataques pueden explotar el inicio de sesión único (SSO), un método ampliamente utilizado que permite a los usuarios acceder a varios servicios con las credenciales de un solo sitio.

Protección y Prevención:

Validación y Seguridad del Código: Validar entradas, prevenir inyecciones SQL y XSS, y realizar revisiones y pruebas de código regulares.

Gestión de Tokens y Subdominios: Usar tokens con restricciones y auditar subdominios antiguos o vulnerables para desactivarlos o actualizarlos.

Autenticación y Protección de Sesiones: Implementar autenticación multifactor (MFA), monitorear inicios de sesión y prevenir redirecciones no autorizadas con listas blancas de URLs.

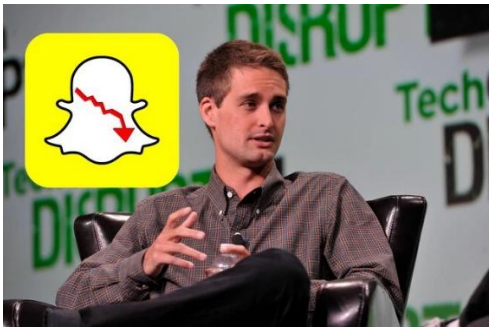
Pruebas de Seguridad: Realizar pruebas de penetración, auditorías periódicas y programas de recompensas por vulnerabilidades para detectar errores.

Capacitación y Monitoreo: Capacitar a desarrolladores en programación segura, educar a usuarios sobre enlaces sospechosos, y monitorear actividad para identificar y responder rápidamente a amenazas.

Whaling o "Caza de Ballenas"

Según Hern (2017), un ataque de "whaling" o "caza de ballenas" afectó a Snapchat en 2016, cuando los atacantes lograron filtrar datos sobre el salario de los empleados de la empresa tras engañar a uno de los ejecutivos mediante un correo electrónico fraudulento. El ataque fue parte de una estafa dirigida específicamente a la alta dirección de la compañía, lo que les permitió obtener información sensible a través de un correo electrónico que parecía provenir del CEO.

Snapchat fue víctima de un ataque de phishing en el que un atacante se hizo pasar por el CEO Evan Spiegel. El atacante envió un correo electrónico a un empleado de recursos humanos solicitando información confidencial sobre los salarios de empleados actuales y pasados, incluyendo detalles sobre opciones de acciones y formularios W-2.



Aunque los datos de los usuarios de Snapchat no fueron comprometidos, los empleados, tanto actuales como antiguos, sí se vieron afectados por la filtración de su información. El ataque consistió en un correo electrónico de phishing, comúnmente utilizado, que parecía provenir del propio CEO de la empresa. En este caso, el estafador se hizo pasar por Evan Spiegel, quien solicitó al empleado de Recursos Humanos información confidencial, la cual fue entregada sin sospechas. Esto demuestra lo fácil que es caer en tales engaños, ya que la solicitud parecía legítima y provenía de un alto ejecutivo. A pesar de la gravedad del ataque, Snapchat tuvo suerte en comparación con otras empresas que han sufrido pérdidas mucho mayores, como Ubiquiti Networks, que envió 46,7 millones de dólares en un ataque similar. Además, se estima que en los últimos dos años y medio, las estafas de este tipo han causado pérdidas por un total de 2.000 millones de dólares.

Protección y Prevención:

Concienciación: Capacitar a los altos ejecutivos sobre las tácticas utilizadas en ataques de whaling y cómo identificarlas.

Verificación de solicitudes: Implementar procedimientos para verificar la autenticidad de solicitudes de transferencias financieras o divulgación de información sensible.

Seguridad del correo electrónico: Utilizar filtros avanzados y autenticación multifactor para proteger las cuentas de correo electrónico de ejecutivos.

Bibliografías

Lindrea, B. (2024, December 30). Hackers que roban cripto dieron un nuevo giro a la estafa de falso empleo para infectar malware. Cointelegraph. https://es.cointelegraph.com/news/crypto-malware-attack-linkedin-job-offer

García, B. (2018, May 7). I LOVE YOU: el virus más famoso del mundo ya es mayor de edad. Blogthinkbig.com. https://blogthinkbig.com/virus-informatico-i-love-you

Zeus: El malware troyano. (2017, December 7). /. https://latam.kaspersky.com/resource-center/threats/zeus-virus?srsltid=AfmBOooRQSiBqzKEsHljO5bBjmhC7hTX_NjLy6h-Xfd5GTrVFUkXCm1V

Jiménez, J. (2022, May 2). “Tenemos la absoluta seguridad de que es un ataque externo”. Los teléfonos de Pedro Sánchez y la. . . Xataka. https://www.xataka.com/legislacion-y-derechos/tenemos-absoluta-seguridad-que-ataque-externo-telefonos-pedro-sanchez-ministra-defensa-han-sido-infectados-pegasus-gobierno

Ruiz, J. J. (2019, March 14). SimBad, el malware para Android presente en más de 200 apps en Google Play. Una Al Día. https://unaaldia.hispasec.com/2019/03/simbad-el-malware-para-android-presente-en-mas-de-200-apps-en-google-play.html

RobbinHood ransomware banks on bad reputation to extort money from victims. (n.d.). https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/robbinhood-ransomware-banks-on-bad-reputation-to-extort-money-from-victims

Destape, E. (2022, January 18). Los 10 casos de ciberdelitos más impactantes de la historia. El Destape. https://www.eldestapeweb.com/informacion-general/ciberdelitos/los-10-casos-de-ciberdelitos-mas-impactantes-de-la-historia-20221180120

es Tusk, la nueva campaña de phishing que quiere robar tu información y criptomonedas. Cinco Días. https://cincodias.elpais.com/smartlife/lifestyle/2024-08-28/tusk-robo-informacion-criptomonedas.html?utm

ChatGPT outages. BleepingComputer. https://www.bleepingcomputer.com/news/security/openai-confirms-ddos-attacks-behind-ongoing-chatgpt-outages/

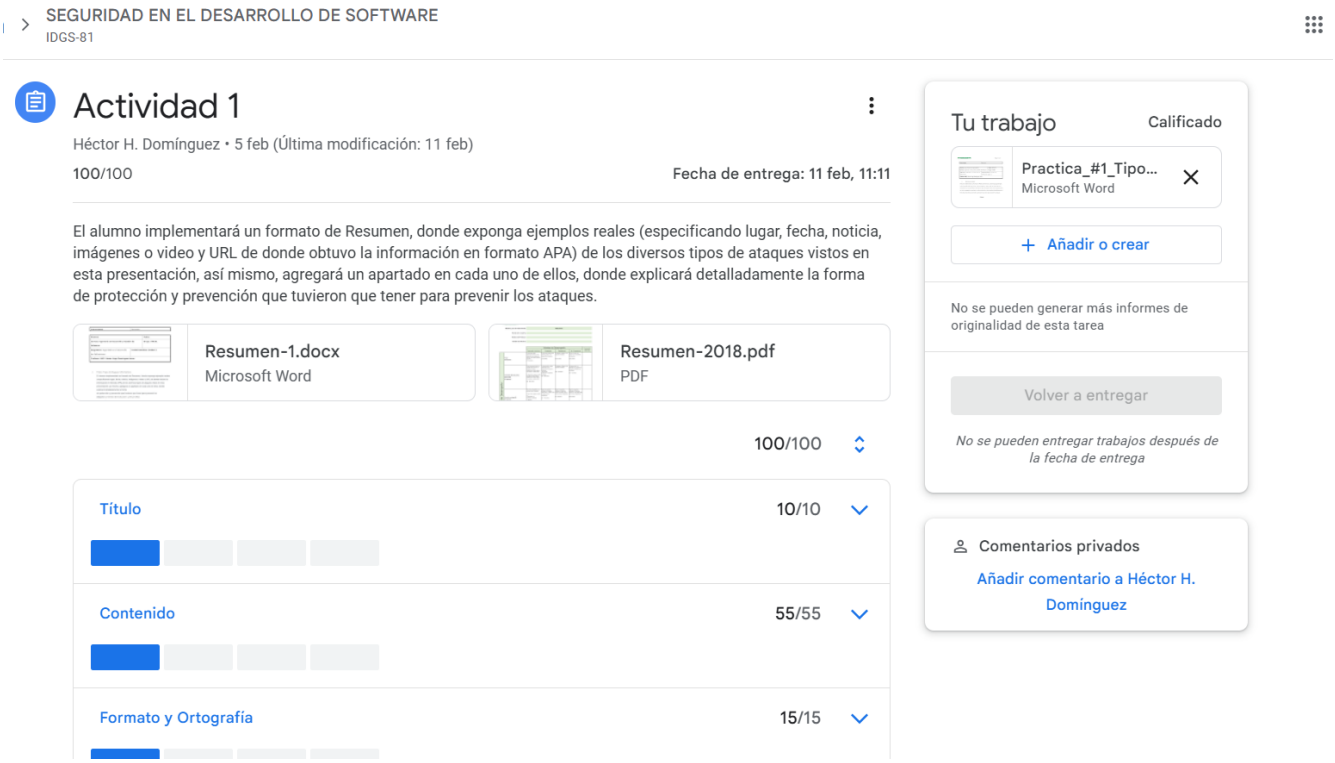
Franceschi-Bicchierai, L., & Franceschi-Bicchierai, L. (2024, July 27). Bugs on Epic Games site allowed hackers to login to any ‘Fortnite’ player’s account. VICE. https://www.vice.com/en/article/fornite-login-hack-epic-games-website/

IBM. (2023, April 25). *Caza de ballenas*. Ibm.com. https://www.ibm.com/es-es/topics/whale-phishing#:~:text=el%20siguiente%20paso-,%C2%BFQu%C3%A9%20es%20la%20caza%20de%20ballenas%3F,texto%20o%20llamadas%20telef%C3%B3nicas%20fraudulentos.

¿Qué es la inyección SQL? ¿Cómo prevenir la inyección de SQL? | Fortinet. (2023). Fortinet. https://www.fortinet.com/lat/resources/cyberglossary/sql-

injection#:~:text=Una%20inyecci%C3%B3n%20SQL%20(SQLi)%20es,nombre%20de%20usuario%20o%20contrase%C3%B1a.

Rubrica



Examen

✓ **Hecho:** Recibir una calificación

Abrió: lunes, 24 de febrero de 2025, 12:20
Cerró: miércoles, 26 de febrero de 2025, 23:59

Para saber como están los conceptos en el alumno

Intentos permitidos: 2

Límite de tiempo: 45 minutos

Método de calificación: Calificación más alta

Calificación para aprobar: 8,0 de 10,0

Resumen de sus intentos previos

Intento	Estado	Revisión
1	Finalizado Enviado: lunes, 24 de febrero de 2025, 23:11	Revisión no permitida
2	Finalizado Enviado: martes, 25 de febrero de 2025, 22:14	Revisión no permitida

Resumen #2_Estándares de Codificación

Instrumento	*Resumen*

Alumno: Jesús Alonso Cornejo Galindo .	**Fecha:** 13/02/25
Carrera: Ingeniería en Desarrollo y Gestión de Software	**Grupo:** IDGS81
Asignatura: Seguridad en el Desarrollo de Aplicaciones	**Unidad temática:** Principios de codificación segura 1
Profesor: MCE. Héctor Hugo Domínguez Jaime	

En un formato de práctica de ejercicios, el alumno realizará una investigación más detallada sobre cada uno de los Estándares de Codificación mencionados en esta presentación. Agregue ejemplos prácticos, captura de imágenes, códigos de ejemplo, etc. No olvide agregar la bibliografía en formato APA.

Titulo

"Estándares de Codificación en Lenguajes de Programación"

I. Introducción:

En este resumen, vamos a desarrollar los estándares de codificación más relevantes de diferentes lenguajes de programación, como JavaScript, PHP y Ruby, entre otros, y su importancia para mejorar la calidad y legibilidad del código. Cada lenguaje tiene sus propias convenciones, como la notación camelCase en JavaScript, el uso adecuado de los espacios de nombres en PHP, y la implementación del patrón de objetos de servicio en Ruby. Estas convenciones facilitan la estructuración del código, mejoran la colaboración entre desarrolladores y aseguran que el código sea fácil de entender y mantener.

Además, vamos a ver lo esencial que es entender lo que no se debe hacer en el código para evitar malas prácticas. En JavaScript, por ejemplo, se debe evitar la declaración de múltiples variables en una sola línea, ya que esto puede hacer que el código sea más difícil de leer y mantener. En PHP, los archivos deben estar centrados en declarar clases y funciones, sin ejecutar lógica adicional que cause efectos secundarios, como modificar configuraciones o generar salidas. En Ruby, se debe evitar el uso de nombres ambiguos o poco descriptivos para funciones y clases, ya que esto puede dificultar la comprensión del propósito del código.

También se abordará la importancia de la validación del código, con herramientas que permiten detectar automáticamente errores o problemas de formato, lo cual asegura que el código cumpla con los estándares establecidos. A través de este análisis, se demostrará cómo seguir los estándares adecuados no solo mejora la calidad del código, sino que también favorece su mantenimiento de tu proyecto.

II. Desarrollo:

1. Estándares de Codificación en Java

De acuerdo con las convenciones de codificación en Java descritas por diversas fuentes, se recomienda que las excepciones nunca sean ignoradas, sino gestionadas de manera adecuada o propagadas cuando sea necesario, evitando el uso de excepciones generales como Exception o Throwable y optando por excepciones más específicas (Gorkovenko, 2024).

```
try {
    int resultado = dividir(a, b);
} catch (ArithmeticException e) {
    System.out.println("Error: División por cero.");
}
```

Según la página web cursa , nos dice que las buenas prácticas indican que cada línea debe contener una sola declaración, inicializando las variables locales en el momento de su declaración y ubicando estas declaraciones al inicio de cada bloque de código.

```
int contador = 0;
String nombre = "Juan";
```

En cuanto a las convenciones de nomenclatura, los paquetes deben nombrarse en minúsculas y, si tienen múltiples palabras, separarlas con puntos para mayor claridad. Además, para las variables temporales utilizadas en bucles, es habitual emplear nombres simples como i, j o k, especialmente cuando se trata de índices de enteros (*Estándares De Codificación De Java*, 2021).

Ejemplo: - para (int i = 0; i <nTables; i ++) {}

2. Estándares de Codificación en C#

Según el autor Sonego (2022), Además, existe debate sobre el uso del guion bajo (_) en la declaración de miembros privados. Algunos desarrolladores lo consideran útil, especialmente al utilizar inyección de dependencias, ya que ayuda a distinguir entre la variable pasada en el constructor y el miembro al cual se asigna. Por ejemplo, al asignar una variable productName en el constructor, se puede utilizar _productName para el miembro privado, evitando confusiones. Aunque el uso de this._ puede parecer tedioso, facilita la lectura y previene el uso incorrecto de variables. Por ejemplo:

```
public class Product
{
    private string _productName;

    public void Show(string productName = null)
    {
        Console.WriteLine($"Mostrar: {productName ?? _productName}!");
    }
}
```

Según el sitio web withoutdebugger, menciona el uso de la palabra clave var en C# es especialmente beneficioso para variables de corta duración o dentro de un mismo ámbito. Esta práctica ayuda a reducir la redundancia y aporta mayor flexibilidad al código. Al declarar una variable con var, el compilador infiere su tipo basándose en el valor asignado, lo que simplifica la escritura y lectura del código. Por ejemplo, en la siguiente declaración:

Es importante utilizar var de manera coherente y consciente, asegurándose de que la inferencia de tipos no comprometa la claridad del código. Aunque var puede simplificar la sintaxis, su uso indiscriminado podría dificultar la comprensión del código por parte de otros desarrolladores o incluso del propio autor en el futuro. Por lo tanto, es recomendable emplearlo cuando el tipo es evidente o irrelevante para la comprensión del contexto, y evitarlo en situaciones donde el tipo de la variable es esencial para la legibilidad y mantenimiento del código (Sonego,2022).

```
var product = new product(id);

var result = await _someApiproduct.GetAsync(id, cancellationToken);

for(var i = 0; i < max; i++)
{
   // i is implicitly typed
```

```
}

var product = this.productFactory.GetOrCreate(id);

this.someDependency.Submitproduct(product);
```

3. Estándares de Codificación en Python

De acuerdo con el autor Navarro, L. (2019), en su guía de estilos para el código de Python, se recomienda utilizar 4 espacios por cada nivel de indentación. Además, las líneas de continuación deben alinear verticalmente los elementos ajustados o emplear una indentación adicional cuando se utilicen paréntesis, corchetes o llaves. Es importante evitar colocar argumentos en la primera línea y usar un nivel extra de indentación para distinguir claramente las líneas de continuación (Navarro, L., 2019, El Solitario).

```
# Alineado con el delimitador de apertura.
foo = long_function_name(var_one, var_two,
                         var_three, var_four)

# Agregar 4 espacios adicionales para distinguir argumentos.
def long_function_name(
        var_one, var_two, var_three,
        var_four):
    print(var_one)

# Uso de indentación colgante.
foo = long_function_name(
    var_one, var_two,
    var_three, var_four)
```

De acuerdo con los Estándares de codificación Python y uso de AutoPEP8 (n.d.), se sugiere que las importaciones de módulos en Python se realicen al inicio del archivo, ubicándolas después del comentario o docstring del módulo y antes de declarar cualquier variable o constante global. El orden recomendado para las importaciones es:

1. Bibliotecas estándar de Python.
2. Bibliotecas de terceros.
3. Bibliotecas de la aplicación local.

```python
# Recomendar
import os
import requests
from xgboost.sklearn import XGBClassifier
from xx.xxx import test  # Importación de aplicaciones locales
# No recomendado
import requests,os  # Debe dividirse en 2 importaciones
from requests import *  # Tabú
```

No se recomienda agrupar múltiples importaciones en una sola línea, como import requests, os, ni usar from requests import *, ya que esto reduce la claridad del código (Estándares de codificación Python, n.d.).

4. Estándares de Codificación en JavaScript

Granada (n.d.) establece que al declarar múltiples variables en JavaScript, todas deben ser declaradas en una sola línea utilizando la misma sentencia var, separadas por comas, con excepción de la última variable, para la cual se usará un punto y coma al final de la línea. Además, a partir de la segunda línea, es recomendable indentar con cuatro espacios para mejorar la legibilidad del código. Las variables deben ser ordenadas alfabéticamente siempre que sea posible. Un ejemplo de declaración incorrecta sería tener variables separadas por saltos de línea sin seguir esta convención. Un ejemplo correcto de declaración sería utilizar comas para separar las variables en una sola línea, y un punto y coma solo al final de la última variable.

- Una declaración no valida:

```
var bad1 = 1;

var iAmFine = false;

var sampleVariable = "wrong";
```

- Una declaración valida:

```
var isHappy = true, // Separa con coma

    score = 100,    // Separa con coma

    isComplete = false; // Termina con punto y coma porque es la última variable
```

Menciona Granada (n.d.) destaca que, en las funciones son fundamentales, consideradas como "ciudadanos de primer tipo". Para declararlas correctamente, es importante seguir ciertas pautas:

1. **Nombres de funciones**: Los nombres de las funciones deben usar la notación camelCase. Un ejemplo sería: function testFunction() {}.

```
function testFunction() {

  // code here

}
```

2. **Llaves y espaciado de funciones**: Las llaves de una función deben comenzar en la misma línea en la que se declara la función, y debe haber un espacio entre el paréntesis de cierre de los argumentos y la llave de inicio del cuerpo de la función.

```
// A. Declaración de función

// B. Parentesis de argumentos

// C. Espacio

// D. Llave de inicio de cuerpo de función.



// ------- A ------- --- B --- C D

function testFunction(username) {

  console.log('Hello ' + username + '...');

}
```

También en este github nos menciona que en JavaScript, las funciones pueden declararse de manera tradicional o asignarse a variables. Cuando una función es asignada a una variable, se debe utilizar un punto y coma (;) al final de la sentencia.

```
function testFunction() {
    console.log('Hello Test...');
}
var anotherTestFunction = function() {
    console.log('Hello Test...');
}; // ; para fin de sentencia.
```

Agregándole tambien los argumentos deben tener nombres en camelCase. Si hay varios, deben separarse por coma y un espacio. Si hay solo un argumento, no se deben dejar espacios.

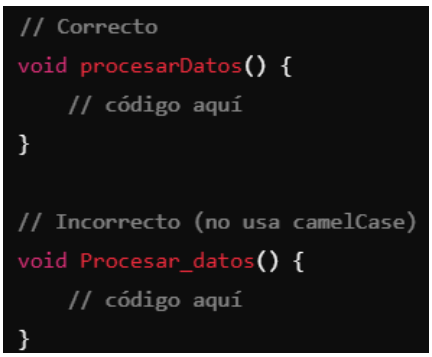
Ejemplo con varios argumentos:

```
function testFunction(arg1, arg2, arg3) {
    console.log('Hello Test...');
}

var testFunction = function(arg1, arg2) {
```

```
    console.log('Hello Test...');

}; // ; para fin de sentencia.
```

5. Estándares de Codificación en C++

Según las normas de programación en C/C++ (n.d.), se recomienda evitar el uso de la notación húngara en el código, ya que fomenta malas prácticas y va en contra de la idea de que el programador debe centrarse en las abstracciones. Cada conjunto de valores debe manejarse con un tipo específico, permitiendo abstraer al programador de los detalles de la implementación. También se sugiere utilizar los tipos nativos del lenguaje, como bool en lugar de macros como BOOL o BOOLEAN. Las variables booleanas deben tener nombres que reflejen su naturaleza lógica, como predicados. Además, las funciones y variables globales deben tener nombres descriptivos, incluso si son largos, mientras que las variables de corto alcance, como contadores, deben ser más simples y concisas (*Normas de Programaci N En C/C++*, 2025).



```
// Correcto
void procesarDatos() {
    // código aquí
}

// Incorrecto (no usa camelCase)
void Procesar_datos() {
    // código aquí
}
```

Se destaca la importancia de usar enum en lugar de macros del preprocesador, ya que ofrecen un mejor control de tipos y son compatibles con los depuradores. Además, la documentación de las funciones debe seguir un formato estandarizado, describiendo su propósito, parámetros y valor de retorno, sin entrar en detalles innecesarios sobre la implementación interna.

```
// Correcto: Usando enum en lugar de una macro

enum Estado {EN_ESPERA, PROCESANDO, TERMINADO};
```

```
void procesar(Estado estado) {
    if (estado == EN_ESPERA) {
        // Código para estado EN_ESPERA
    }
}

// Incorrecto: Usando una macro para definir los estados
#define EN_ESPERA 0
#define PROCESANDO 1
#define TERMINADO 2

void procesar(int estado) {
    if (estado == EN_ESPERA) {
        // Código para estado EN_ESPERA
    }
}
```

En cuanto a las buenas prácticas generales, el código no debe generar advertencias al compilarse, y todas las funciones en C deben tener prototipos. Además, las funciones cuyo ámbito esté limitado al archivo en el que se definen deben ser declaradas como static para evitar conflictos de nombres y mejorar la eficiencia. Se prefiere que las funciones sean breves y se utilicen para encapsular unidades funcionales, facilitando la legibilidad y el mantenimiento del código. También se recomienda devolver directamente el resultado de una función en lugar de modificar parámetros pasados por referencia.

```
// Correcto: función estática (solo utilizada dentro del archivo)
static void realizarOperacion() {
    // Código aquí
}

// Incorrecto: función global sin necesidad de serlo
void realizarOperacion() {
    // Código aquí
}
```

6. Estándares de Codificación en Ruby

Según el artículo "Patrón Service object en Ruby on Rails" de CódigoFacilito, se recomienda utilizar el patrón de diseño Service Object para encapsular y reutilizar la lógica de negocio compleja en aplicaciones Ruby on Rails. Este patrón ayuda a mantener el código limpio y bien estructurado, evitando que modelos y controladores implementen lógica de negocio que no les corresponde.

Ejemplo de implementación del patrón Service Object:

Imaginemos que estamos desarrollando una tienda en línea y necesitamos implementar el proceso de checkout, que incluye los siguientes pasos:

1. Validar la orden.
2. Completar y confirmar el proceso de pago.
3. Crear un registro de pago exitoso.
4. Enviar un correo electrónico de confirmación de compra.
5. Actualizar el inventario de los productos adquiridos.

En lugar de manejar toda esta lógica en el controlador, podemos crear un Service Object para encapsularla:

```
# app/services/order_checkout_service.rb

class OrderCheckoutService

  def initialize(order)

    @order = order

  end
```

```
def call
  validate_order
  process_payment
  create_payment_record
  send_confirmation_email
  update_inventory
  clear_cart
end

private

def validate_order
  # Lógica para validar la orden
end

def process_payment
  # Lógica para procesar el pago
```

```
end

def create_payment_record
  # Lógica para crear el registro de pago
end

def send_confirmation_email
  # Lógica para enviar el correo de confirmación
end

def update_inventory
  # Lógica para actualizar el inventario
end

def clear_cart
  # Lógica para limpiar el carrito de compras
end
```

```
end
```

Luego, en el controlador, simplemente llamamos al Service Object:

```
# app/controllers/orders_controller.rb

class OrdersController < ApplicationController

  def create

    @order = Order.new(order_params)

    OrderCheckoutService.new(@order).call

    flash[:success] = "¡Pedido realizado con éxito!"

    redirect_to root_path

  end

end
```

Este enfoque mejora la legibilidad y mantenibilidad del código, delegando la responsabilidad de manejar el proceso de checkout al Service Object (*Patrón Service Object En Ruby on Rails - Blog De Código Facilito*, n.d.).

7. Estándares de Codificación en PHP

De acuerdo con el sitio web coppeldev (*PHP - Codificación Estándar Básica*, n.d.), un estándar es que el código debería limitarse a declarar estructuras como clases, funciones y constantes, sin causar efectos secundarios. Esto significa que la lógica de negocio no debe ejecutarse en la misma fase que la declaración de estas estructuras, sino que debería separarse en diferentes partes del código. Los "efectos secundarios" como requiere e include incluyen actividades como la generación de salida, la inclusión de archivos, la conexión a servicios externos, la modificación de configuraciones, el envío de errores o excepciones, la manipulación de variables globales o estáticas, y la lectura o escritura de archivos.

Este es un ejemplo que puede seguir, sin efectos secundarios:

```
<?php

// declaración

function foo()

{

    // cuerpo de la función

}


// una declaración condicional *no* es un

// efecto secundario
```

```
if (! function_exists('bar')) {
    function bar()
    {
        // cuerpo de la función
    }
}
```

Este ejemplo incluye efectos secundarios y es lo que debe evitar:

```
<?php
// efecto secundario: cambiar configuracion inicial
ini_set('error_reporting', E_ALL);

// efecto secundario: cargar ficheros
include "archivo.php";

// efecto secundario: generar salida
echo "<html>\n";
```

```
// declaración

function foo()

{

    // cuerpo de la función

}
```

Por ejemplo, un archivo que declare funciones y, al mismo tiempo, cause efectos secundarios (como cambiar configuraciones, cargar archivos o generar salida) no sigue la práctica recomendada. En cambio, se recomienda que los archivos contengan solo declaraciones y que cualquier ejecución lógica se realice de manera controlada y separada.

Respecto a los espacios de nombres y las clases, se deben seguir las normas PSR-0. Esto implica que cada clase debe estar en un archivo independiente y ser parte de un espacio de nombres, con al menos un nivel de jerarquía para el nombre del proveedor. Los nombres de las clases deben emplear la notación *StudlyCaps*, y el código en PHP 5.3 o superior debe usar espacios de nombres de manera formal.

```
<?php

// PHP 5.3 o superior:

namespace Proveedor\Modelo;


class Foo
```

{

}

Tambien este sitio web nos guía a dar recomendaciones específicas sobre el uso de notaciones como $StudlyCaps, $camelCase o $guion_bajo para nombrar las propiedades. Sin embargo, sea cual sea la convención de nomenclatura elegida, debe aplicarse de manera coherente dentro de un alcance razonable, que puede ser a nivel de proveedor, paquete, clase o método.

8. Estándares de Desarrollo Web en HTML/CSS

Según el sitio web mosaic.uoc, HTML y XHTML son lenguajes de marcado que se componen de elementos con atributos, algunos de los cuales son opcionales y otros obligatorios. Estos elementos se utilizan para identificar los diferentes tipos de contenido dentro de un documento, indicándole a los navegadores web cómo deben presentarlo (por ejemplo, encabezados, párrafos, tablas, listas con viñetas, entre otros). Los elementos determinan la naturaleza del contenido, mientras que los atributos añaden información extra, como una ID para su identificación o un enlace a una ubicación. Es importante que el etiquetado sea lo más semántico posible, describiendo la función del contenido de manera clara y precisa (De Catalunya, n.d.).

Figura 1. Anatomía de un elemento (X)HTML.

De acuerdo con el sitio al articulo introducción al mundo de los estándares web, HTML, XHTML y CSS son estándares establecidos, y el World Wide Web Consortium (W3C) ha desarrollado una herramienta llamada validador. Esta herramienta revisa automáticamente las páginas web y detecta errores en el código, como etiquetas de cierre faltantes o la ausencia de comillas alrededor de los atributos (De Catalunya, n.d.).

```
<html>

 <head>

   <title>Ejemplo de error</title>

 </head>
```

```
<body>
  <h1>Bienvenidos a mi página</h1>
  <p>Este es un párrafo cerrado correctamente</p>
  <a href="https://www.ejemplo.com">Enlace de ejemplo</a>
</body>
</html>
```

Al validar el código a través del validador de W3C, este marcaría los errores y sugeriría las correcciones necesarias para que el código cumpla con los estándares.

III. Bibliografías:

Mejores prácticas y estándares de codificación de Java : Curso Aprende a programar en Java completo, desde programación lógica hasta avanzada | Cursa. (n.d.). Cursa.

https://cursa.app/es/pagina/mejores-practicas-y-estandares-de-codificacion-de-java

Gorkovenko, A. (2024, February 2). *Convenciones de codificación de Java. Cuáles seguir y por qué*. CodeGym. https://codegym.cc/es/groups/posts/es.491.convenciones-de-codificacion-de-java-cuales-seguir-y-por-que

Estándares de codificación de Java. (2021, June 7). ICHI.PRO. https://ichi.pro/es/estandares-de-codificacion-de-java-115149889496980

Navarro, L. (2019, June 14). *Guía de estilos para el código de Python*. El Solitario. https://elsolitario.org/2019/01/30/guia-de-estilos-para-el-codigo-de-python/

Estándares de codificación Python y uso de AutoPEP8 - programador clic. (n.d.). https://programmerclick.com/article/23781149295/

Granada. (n.d.). *GitHub - 0granada/js-coding-standards: Estandares de codificación para el lenguaje javascript*. GitHub. https://github.com/0granada/js-coding-standards

Normas de programación en C/C++. (n.d.). http://www.reloco.com.ar/prog/normas.html

Patrón Service object en Ruby on Rails - Blog de Código Facilito. (n.d.). CódigoFacilito. https://codigofacilito.com/articulos/service-object-ruby-on-rails

PHP - Codificación estándar básica. (n.d.). https://coppeldev.github.io/php/standards/psr-1.html

de, O. (2017). *Introducción al mundo de los estándares web - El modelo de estándares web: HTML, CSS y JavaScript*. Uoc.edu. https://mosaic.uoc.edu/ac/le/es/m1/ud3/index.html

Rubrica


Actividad 2
Héctor H. Domínguez • 5 feb (Última modificación: 5 feb)
100 puntos
Fecha de entrega: 15 feb, 0:59
En un
formato de Resumen, el alumno realizará una investigación más detallada sobre cada uno de los Estándares de Codificación mencionados en esta presentación. Agregue ejemplos prácticos, captura de imágenes, códigos de ejemplo, etc. No olvide agregar la bibliografía en formato APA.
Resumen.docx
Resumen-2018.pdf
PDF
/100
Título
/10
Contenido
/55
Formato y Ortografía
/15
Tu trabajo
Entregado
Resumen #2_Estándar...
Microsoft Word
Anular entrega
No se pueden entregar trabajos después de la fecha de entrega
Comentarios privados
Añadir comentario a Héctor H. Domínguez

Unidad 2

Resumen #1 Frameworks del Frontend y del Backend

Instrumento	*Resumen*

Alumno: Jesús Alonso Cornejo Galindo	**Fecha:** 18/03/25
Carrera: Ingeniería en Desarrollo y Gestión de Software	**Grupo:** IDGS81
Asignatura: Seguridad en el Desarrollo de Aplicaciones	**Unidad temática:** Frameworks y Seguridad
Profesor: MCE. Héctor Hugo Domínguez Jaime	

El alumno realizará una investigación sobre los diversos Frameworks que existen para el desarrollo de aplicaciones web, tanto para el uso del Fronted como para el uso en el Backend.
Organícelos por categoría (fronted y backend), descríbalos, Mencione sus ventajas y desventajas y muy importante mencione en que tipo de ocasión se debe de usar cada uno de ellos.
No olvide especificar la bibliografía en formato APA.

Titulo

" Frameworks del Frontend y del Backend"

Introducción

En la actualidad, el desarrollo de aplicaciones web y móviles ha evolucionado considerablemente gracias a una amplia variedad de frameworks y bibliotecas disponibles, cada uno con sus características únicas, ventajas y limitaciones. Estas herramientas permiten a los desarrolladores crear aplicaciones más eficientes, escalables y fáciles de mantener. Frameworks como **React.js**, **Angular** y **Vue.js** dominan el desarrollo frontend, ofreciendo enfoques innovadores para la construcción de interfaces interactivas y modulares, mientras que frameworks backend como **Django**, **Express.js** y **Laravel** proporcionan soluciones robustas para gestionar la lógica de negocio y la interacción con bases de datos. Además, tecnologías como **Spring Boot** y **Ruby on Rails** facilitan la creación de aplicaciones empresariales escalables y seguras. Cada uno de estos frameworks tiene un enfoque diferente en cuanto a rendimiento, facilidad de uso y la estructura del código, lo que permite que los desarrolladores elijan la herramienta adecuada en función de las necesidades específicas del proyecto. Con la continua evolución del desarrollo web, elegir el marco correcto se ha vuelto fundamental para garantizar aplicaciones de alta calidad, fáciles de mantener y con una experiencia de usuario óptima.

IV. Desarrollo:

Frameworks de Frontend

"Los frameworks de frontend se utilizan para diseñar y desarrollar la interfaz de usuario de una aplicación web. Estos frameworks proporcionan herramientas que permiten construir experiencias interactivas y dinámicas con mayor facilidad y eficiencia" (Haverbeke, M. (2018). *Eloquent JavaScript: A Modern Introduction to Programming*).

React.js

Descripción

React.js es una biblioteca de JavaScript desarrollada por Facebook en 2013 para la creación de interfaces de usuario basadas en componentes reutilizables. Según Facebook (2013), React permite actualizar eficientemente la interfaz gracias a su sistema de Virtual DOM, que optimiza las actualizaciones del DOM real, mejorando el rendimiento y la experiencia del usuario.

Ventajas

- **Modularidad y reutilización de componentes:** Permite crear componentes independientes que pueden ser reutilizados a lo largo de la aplicación, lo que mejora la mantenibilidad del código.
- **Rápida renderización gracias al Virtual DOM:** El Virtual DOM minimiza las actualizaciones al DOM real, lo que reduce el costo computacional y mejora la experiencia del usuario.
- **Amplia comunidad y ecosistema de herramientas:** React cuenta con una enorme comunidad y un ecosistema rico en herramientas, bibliotecas y plugins, lo que facilita la integración de nuevas funcionalidades y la resolución de problemas comunes.

Desventajas

- **Curva de aprendizaje empinada para principiantes:** Aunque la base de React es simple, el ecosistema completo puede resultar desafiante, especialmente para aquellos sin experiencia previa en el desarrollo de interfaces dinámicas.
- **Requiere configuraciones adicionales para proyectos grandes:** A medida que los proyectos crecen, React puede necesitar configuraciones avanzadas y herramientas adicionales como Redux, React Router o Webpack para gestionar el estado y las rutas, lo que puede aumentar la complejidad.

Cuándo usarlo

- **Aplicaciones web interactivas con gran cantidad de actualizaciones de UI:** React es ideal para aplicaciones que requieren una actualización constante del estado de la interfaz, como redes sociales o aplicaciones de colaboración en tiempo real.
- **Proyectos que requieren una arquitectura basada en componentes reutilizables:** Si el proyecto demanda una organización modular y una reutilización de código eficiente, React es una opción excelente.

Angular

Descripción

Angular es un framework desarrollado por Google en 2010. De acuerdo con Google (2010), Angular implementa una arquitectura basada en MVC (Modelo-Vista-Controlador), lo que facilita la organización del código y fomenta buenas prácticas de desarrollo. Angular también es conocido por ser un framework completo, que proporciona soluciones integradas para la gestión de rutas, formularios, validación y más.

Ventajas

- **Arquitectura bien estructurada con MVC:** La implementación de MVC en Angular ayuda a separar las preocupaciones en la aplicación, haciendo que el código sea más mantenible y escalable.

- **Uso de TypeScript para mayor seguridad en el código:** Angular se basa en TypeScript, que agrega tipado estático y otras características que ayudan a detectar errores antes de que ocurran durante la ejecución.
- **Soporte oficial de Google:** Angular es respaldado por Google, lo que le otorga una alta estabilidad y mantenimiento a largo plazo.

Desventajas

- **Complejidad en su curva de aprendizaje:** La riqueza del framework y su amplio conjunto de características pueden hacer que Angular sea más difícil de aprender, especialmente para quienes no están familiarizados con la programación orientada a objetos o TypeScript.
- **Puede ser pesado para aplicaciones pequeñas:** Debido a su robustez, Angular puede resultar excesivo para aplicaciones pequeñas o de bajo rendimiento, donde un enfoque más simple como React o Vue.js podría ser más adecuado.

Cuándo usarlo

- **Aplicaciones empresariales escalables:** Angular es ideal para grandes aplicaciones empresariales que requieren una arquitectura bien definida, y donde la modularidad y la seguridad son fundamentales.
- **Proyectos con requerimientos estrictos de seguridad y modularidad:** Su robusta estructura y su enfoque en la seguridad hacen que Angular sea una opción preferida para aplicaciones con necesidades de alto nivel de seguridad y control.

Vue.js

Descripción

Vue.js es un framework progresivo de JavaScript creado por Evan You en 2014. You (2014) destaca que Vue.js es fácil de integrar y aprender, lo que lo convierte en una opción atractiva para nuevos desarrolladores. A

diferencia de Angular, Vue no es un framework completo, sino que se centra principalmente en la capa de vista de una aplicación, permitiendo a los desarrolladores agregar funcionalidades de manera gradual.

Ventajas

- **Sintaxis sencilla y fácil de aprender:** Vue tiene una sintaxis clara y concisa que facilita la integración en proyectos existentes, y es particularmente amigable para desarrolladores que están comenzando con frameworks JavaScript.
- **Ligero y rápido:** Vue es más ligero que otros frameworks, lo que lo convierte en una excelente opción para aplicaciones pequeñas o medianas donde el rendimiento es una prioridad.
- **Flexible para integrarse con otros proyectos:** Vue puede ser usado como una librería para crear interactividad en una página web o como un framework completo para construir aplicaciones más grandes.

Desventajas

- **Menor comunidad en comparación con React y Angular:** Aunque Vue tiene una comunidad activa, aún es más pequeña que la de React o Angular, lo que puede dificultar encontrar soporte en línea o soluciones a problemas complejos.
- **Menos adoptado en grandes empresas:** Aunque Vue está ganando popularidad, especialmente en startups y pequeñas empresas, su adopción en grandes corporaciones aún es limitada en comparación con React y Angular.

Cuándo usarlo

- **Aplicaciones web ligeras y flexibles:** Vue es ideal para proyectos pequeños o medianos que requieren una solución rápida y eficiente sin una sobrecarga de funcionalidades innecesarias.
- **Proyectos que requieren una rápida implementación:** Si se necesita desarrollar un proyecto con una curva de aprendizaje baja y una rápida integración con otras tecnologías, Vue es una excelente opción.

Svelte

Descripción

Svelte es un framework creado por Rich Harris en 2016. Harris (2016) explica que Svelte no utiliza Virtual DOM, sino que compila los componentes directamente en JavaScript optimizado, lo que resulta en una menor sobrecarga durante la ejecución. Esto permite que las aplicaciones construidas con Svelte sean más rápidas y ligeras en comparación con otros frameworks que dependen de un Virtual DOM.

Ventajas

- **Código altamente optimizado y rápido:** Svelte compila los componentes en código JavaScript puro, lo que reduce significativamente el tamaño de la aplicación y mejora su rendimiento.
- **Sintaxis simple y amigable:** Svelte tiene una sintaxis limpia y fácil de aprender, lo que lo convierte en una excelente opción para desarrolladores nuevos o aquellos que buscan productividad.
- **No depende de un Virtual DOM:** Al no usar Virtual DOM, Svelte minimiza la complejidad y mejora la eficiencia al compilar los componentes directamente en el código que se ejecuta en el navegador.

Desventajas

- **Comunidad y ecosistema más pequeños:** Aunque Svelte ha ganado popularidad rápidamente, su comunidad sigue siendo más pequeña en comparación con React o Angular, lo que puede dificultar encontrar soporte o soluciones a problemas complejos.
- **Menos soporte empresarial:** A pesar de su eficiencia, Svelte aún no tiene una adopción tan amplia en grandes empresas, lo que limita el soporte y la estabilidad que ofrecen otros frameworks más establecidos.

Cuándo usarlo

- **Aplicaciones con requerimientos de alto rendimiento:** Si el rendimiento es una prioridad, Svelte es ideal debido a su enfoque optimizado y su capacidad para generar código muy eficiente.

- **Proyectos con equipos pequeños que buscan simplicidad:** Svelte es perfecto para equipos pequeños o proyectos donde la simplicidad y rapidez de desarrollo son esenciales, ya que no requiere configuraciones complicadas ni dependencias externas pesadas.

Bootstrap

Descripción

Bootstrap es un framework CSS desarrollado por Twitter en 2011. Según Twitter (2011), Bootstrap permite la creación rápida de diseños responsivos con un sistema de grillas flexible que facilita la creación de interfaces de usuario que se adaptan a diferentes tamaños de pantalla. Bootstrap incluye un conjunto de estilos predefinidos, componentes y utilidades, lo que acelera el desarrollo frontend.

Ventajas

- **Facilidad de uso e implementación:** Bootstrap es fácil de integrar en cualquier proyecto, gracias a su extensa documentación y ejemplos prácticos, lo que permite que los desarrolladores comiencen rápidamente.
- **Diseño responsivo predeterminado:** Al utilizar el sistema de grillas flexible, Bootstrap ofrece un diseño responsivo que se ajusta automáticamente a diferentes tamaños de pantalla, lo que garantiza una experiencia de usuario consistente en dispositivos móviles y de escritorio.
- **Gran compatibilidad con navegadores:** Bootstrap es altamente compatible con los principales navegadores web, lo que minimiza los problemas de visualización entre diferentes plataformas.

Desventajas

- **Puede generar diseños visualmente repetitivos:** Aunque Bootstrap facilita el diseño, el uso extensivo de sus componentes prediseñados puede resultar en aplicaciones con una apariencia similar a otros sitios web que también usan Bootstrap.

- **Puede ser pesado en proyectos complejos:** A medida que las aplicaciones crecen, Bootstrap puede volverse pesado si no se usa de manera eficiente, ya que incluye una gran cantidad de funcionalidades y estilos que tal vez no se necesiten en todos los proyectos.

Cuándo usarlo

- **Proyectos que requieren interfaces rápidas y atractivas:** Si se necesita crear una interfaz atractiva y funcional rápidamente, Bootstrap es ideal por su facilidad de uso y componentes predefinidos.
- **Aplicaciones con necesidad de componentes predefinidos:** Cuando se necesitan componentes de UI comunes como botones, formularios, y alertas, Bootstrap proporciona una solución lista para usar que ahorra tiempo de desarrollo.

Frameworks de Backend

Django

Descripción

Django es un framework de Python creado en 2005. Según Holovaty y Kaplan-Moss (2005), Django sigue el principio de "baterías incluidas", proporcionando múltiples herramientas integradas que permiten a los desarrolladores crear aplicaciones web de manera rápida y con menos esfuerzo. Este enfoque hace que Django sea muy adecuado para desarrollos que requieren autenticación, administración, formularios, entre otros.

Ventajas

- **Seguridad integrada:** Django incluye mecanismos de seguridad integrados, como protección contra inyecciones SQL, CSRF (Cross-Site Request Forgery) y XSS (Cross-Site Scripting), lo que lo convierte en una opción muy segura para el desarrollo web.
- **Desarrollo rápido:** El enfoque "baterías incluidas" permite a los desarrolladores crear aplicaciones funcionales rápidamente, ya que muchas funcionalidades básicas ya están preconfiguradas.
- **Escalabilidad:** Django es escalable, lo que lo hace adecuado tanto para proyectos pequeños como para aplicaciones empresariales de gran tamaño que requieren manejar una gran cantidad de datos o tráfico.

Desventajas

- **Puede ser complejo para principiantes:** Aunque Django facilita muchos aspectos del desarrollo, su estructura y configuraciones pueden resultar complejas para los desarrolladores sin experiencia previa.

- **Requiere más recursos en aplicaciones grandes:** Si bien es escalable, Django puede consumir más recursos en aplicaciones muy grandes debido a su naturaleza monolítica y la cantidad de características que ofrece.

Cuándo usarlo

- **Aplicaciones empresariales con necesidades de seguridad y escalabilidad:** Django es ideal para proyectos grandes que requieren una estructura robusta y necesitan integrar características como autenticación, administración y manejo de base de datos con seguridad avanzada.

Express.js

Descripción

Express.js es un framework de Node.js diseñado para la creación de APIs y aplicaciones web. Según Dahl (2009), Express.js facilita el desarrollo backend con una estructura mínima pero flexible, permitiendo a los desarrolladores añadir solo las características necesarias para su aplicación sin mucha sobrecarga.

Ventajas

- **Ligero y rápido:** Express.js es muy ligero y rápido, lo que lo hace adecuado para aplicaciones que necesitan alta eficiencia y tiempos de respuesta rápidos.
- **Gran compatibilidad con Node.js:** Al ser un framework basado en Node.js, Express aprovecha al máximo las ventajas de Node, como su capacidad para manejar muchas conexiones concurrentes.

Desventajas

- **Menos estructurado que otros frameworks:** A diferencia de otros frameworks como Django o Laravel, Express tiene una estructura más libre y no impone tantas reglas sobre cómo organizar el código, lo que puede llevar a la creación de aplicaciones desordenadas si no se tiene cuidado.

- **Seguridad depende de implementaciones externas:** Express no incluye mecanismos de seguridad tan robustos como otros frameworks, lo que significa que el desarrollador debe integrar medidas de seguridad adicionales.

Cuándo usarlo

- **Desarrollo de APIs REST:** Express es perfecto para crear APIs RESTful debido a su estructura minimalista y su compatibilidad con Node.js.
- **Aplicaciones con microservicios:** Su ligereza y flexibilidad hacen que Express sea adecuado para aplicaciones basadas en microservicios, donde cada servicio puede ser pequeño y autónomo.

Laravel

Descripción

Laravel es un framework de PHP creado por Taylor Otwell en 2011. Otwell (2011) destaca su enfoque en la elegancia del código y la facilidad de desarrollo. Laravel ha sido diseñado para facilitar tareas comunes de desarrollo web, como la autenticación, el manejo de sesiones y las migraciones de bases de datos, lo que mejora la eficiencia del desarrollo.

Ventajas

- **Sistema de migraciones de bases de datos:** Laravel proporciona un sistema de migraciones que permite gestionar de manera eficiente las modificaciones de las bases de datos, facilitando la creación y mantenimiento de estructuras de datos.
- **Arquitectura MVC clara:** Laravel sigue el patrón de diseño MVC (Modelo-Vista-Controlador), lo que facilita la separación de responsabilidades y mejora la mantenibilidad del código.

Desventajas

- **Puede ser pesado para proyectos pequeños:** Laravel puede resultar demasiado complejo y pesado para aplicaciones pequeñas o proyectos que no requieran las características avanzadas que ofrece el framework.
- **Curva de aprendizaje inicial:** Aunque Laravel está bien documentado, su curva de aprendizaje puede ser algo empinada para los desarrolladores que no están familiarizados con su arquitectura o las herramientas asociadas.

Cuándo usarlo

- **Aplicaciones web con una estructura bien definida:** Laravel es ideal para proyectos de tamaño medio a grande que requieren una arquitectura clara y bien definida, así como la integración de funcionalidades avanzadas de backend como autenticación y manejo de bases de datos.

Spring Boot

Descripción

Spring Boot es un framework de Java diseñado para crear aplicaciones empresariales con configuración mínima. Pivotal Software (2014) destaca su compatibilidad con microservicios, lo que permite a los desarrolladores crear aplicaciones modulares y escalables sin necesidad de una configuración extensa.

Ventajas

- **Alto rendimiento:** Spring Boot está optimizado para ofrecer un rendimiento eficiente, lo que lo convierte en una opción ideal para aplicaciones empresariales que necesitan procesar grandes volúmenes de datos.
- **Seguridad avanzada:** Spring Boot incluye mecanismos de seguridad avanzados y se integra fácilmente con Spring Security, lo que proporciona una base robusta para desarrollar aplicaciones seguras.

Desventajas

- **Uso intensivo de memoria:** Aunque Spring Boot es altamente eficiente en términos de funcionalidad, las aplicaciones construidas con este framework pueden consumir una cantidad significativa de memoria, lo que podría ser un problema en entornos con recursos limitados.

Cuándo usarlo

- **Aplicaciones empresariales escalables:** Spring Boot es adecuado para aplicaciones grandes y complejas que requieren escalabilidad y alto rendimiento, como aplicaciones financieras o de gestión de recursos empresariales.

Ruby on Rails

Descripción

"Ruby on Rails es un framework basado en Ruby" según Hansson (2004), Rails permite un desarrollo ágil y estructurado al ofrecer una serie de herramientas que facilitan tareas comunes como la gestión de bases de datos, la autenticación de usuarios y la creación de interfaces web.

Ventajas

- **Desarrollo rápido:** Ruby on Rails favorece el desarrollo rápido gracias a su enfoque en la convención sobre configuración, lo que permite a los desarrolladores escribir menos código para lograr más funcionalidad.
- **Manejo eficiente de bases de datos:** Rails tiene una excelente integración con bases de datos y proporciona herramientas como ActiveRecord para gestionar las interacciones con la base de datos de manera sencilla y eficiente.

Desventajas

- **Requiere conocimientos previos de Ruby:** Para sacar el máximo provecho de Ruby on Rails, es necesario tener experiencia con Ruby, lo que puede ser un obstáculo para los desarrolladores que no están familiarizados con este lenguaje.

Cuándo usarlo

- **Aplicaciones que requieren un desarrollo rápido:** Ruby on Rails es perfecto para proyectos donde el tiempo de desarrollo es crucial y se busca una solución rápida y eficiente para crear aplicaciones web, como startups o MVPs (productos mínimos viables).

V. Bibliografías:

Node.js — The V8 JavaScript Engine. (2015). Nodejs.org. https://nodejs.org/en/learn/getting-started/the-v8-javascript-engine

React – A JavaScript library for building user interfaces. (2021). Reactjs.org. https://legacy.reactjs.org/

Google. (2010). *Angular: A platform for building mobile and desktop web applications*.

https://medium.com/@softradix2022/angular-development-usa-comprehensive-guide-to-angular-framework-01dc935997d2

Hansson, D. H. (2004). *Ruby on Rails: Web application framework*. 37signals.

https://37signals.com/podcast/ruby-on-rails/

Harris, R. (2016). *Svelte: Cybernetically enhanced web apps*. Svelte.dev.

https://dev.to/stackfoss/cybernetically-enhanced-web-apps-svelte-16ge

Holovaty, A., & Kaplan-Moss, J. (2005). *Django: The web framework for perfectionists with deadlines*. Django Software Foundation.

https://www.djangoproject.com/

Otwell, T. (2011). *Laravel: The PHP framework for web artisans*. Laravel LLC.

https://laravel.com/

Pivotal Software. (2014). *Spring Boot: Simplifying Java development*. VMware Inc.

https://www.vmware.com/topics/spring-boot

Twitter. (2011). *Bootstrap: The world's most popular front-end open source toolkit*. Twitter Inc.https://getbootstrap.com/

You, E. (2014). *Vue.js: The progressive JavaScript framework*. Vue.js Organization.

https://vuejs.org/

Rubrica

Actividad 1 Unidad II: Frameworks y Seguridad

✓ Hecho

Apertura: martes, 11 de marzo de 2025, 00:00
Cierre: martes, 18 de marzo de 2025, 23:59

El alumno realizará una investigación sobre los diversos Frameworks que existen para el desarrollo de aplicaciones web, tanto para el uso del Fronted como para el uso en el Backend.
Organícelos por categoría (fronted y backend), descríbalos, Mencione sus ventajas y desventajas y muy importante mencione en que tipo de ocasión se debe de usar cada uno de ellos.
No olvide especificar la bibliografía en formato APA.

Editar entrega | Borrar entrega

Estado de la entrega

Estado de la entrega	Enviado para calificar
Estado de la calificación	Sin calificar
Tiempo restante	La tarea fue enviada 1 día 4 horas antes de la fecha límite
Última modificación	lunes, 17 de marzo de 2025, 19:26