TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLALNEPANTLA

INSTITUTO TECNOLÓGICO DE TLALNEPANTLA.

TECNOLOGÍAS DE SEGURIDAD EN SOFTWARE.

DOCENTE: DÍAZ RINCÓN HILDA.

ALUMNO Y N^{o} CONTROL:

ALDAMA MÁRQUEZ JOEL JAIR - 15251870.

GRUPO: T91.

ACTIVIDAD: 3.2 CONTROLES EN APLICACIONES EN DESARROLLO

Contenido

Desarrollo seguro de aplicaciones web.

Seguridad

La seguridad tiene que ver con la protección de los datos críticos y la protección de los atributos de los datos críticos, como:

- Disponibilidad, ya que los datos deben de estar disponibles en el momento en el que se le tiene que utilizar.
- Integridad: El dato debe de estar integro cuando se le recupere y se le visualiza.
- Confidencialidad: Cuando la persona que está autorizada puede acceder a la información.

Estas 3 características del dato es lo que se debe de proteger a través de la seguridad.

Cuando se protegen archivos informáticos es que no es posible proteger todo durante todo el tiempo y de todos los problemas de todos los riesgos y se debe decidir en donde poner más importancia en la protección.

La aplicación en algún momento, por alguna razón va a ser atacada desde algún lugar y eso es un hecho, se ve diariamente.

En la ciberseguridad existe un área en específico llamada CSIRT, que es un Centro de Respuesta a Incidentes Informáticos en el que se procesa información de ataques que reciben los sitios web, los datos se toman de diferentes mecanismos, diferentes ondas de equipamiento informático de terceros.

Dato: Se encontró que un servidor de un organismo estaba emitiendo, hasta tenia encolado 760,000 correos para salir, que eran spam de un servidor comprometido que estaba saliendo hacia el exterior, esto quiere decir, que si tenemos aunque sea un equipo comprometido puede generar un gran número de ataques de spam a diferentes IPs.

Hay una realidad, en algún momento si tenemos una aplicación expuesta en internet va a ser atacada, lo único que faltaría por saber es ¿Cuándo?, para poder prevenir este tipo de ataques debemos de poner en énfasis lo que es el desarrollo de aplicaciones web.

Ataque o protección.

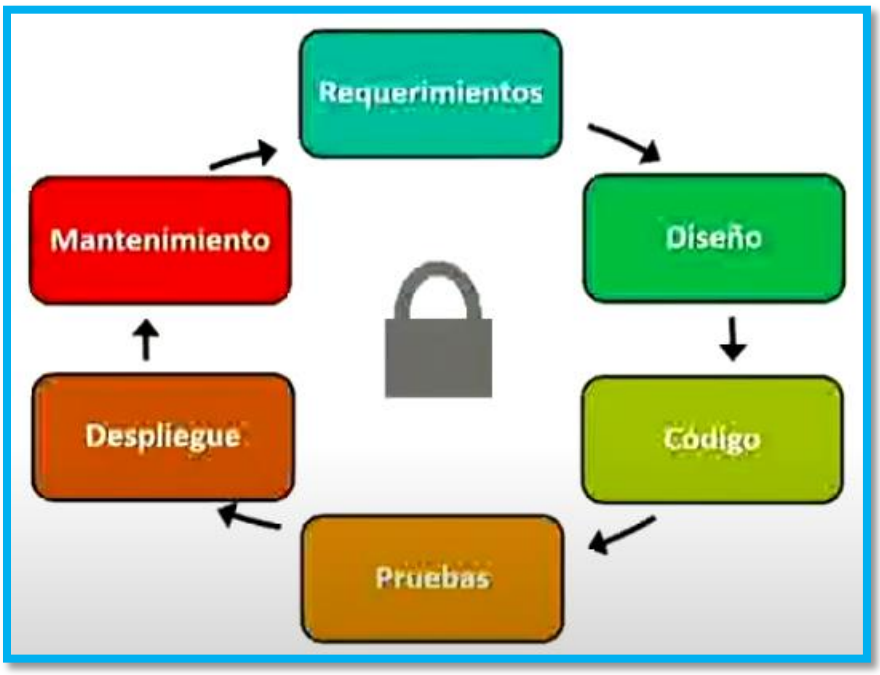
Lo fundamental para los desarrolladores es que sepamos si los datos que tenemos que proteger son críticos o no puede ser quizás una aplicación web que es una página estática donde solamente hay alguna foto y se le hace promoción a un sitio o puede tratarse un sitio web financiero o de una índole más crítica por lo que es fundamental que antes de aplicar cualquier tipo de protección en lo que tiene que ver con el desarrollo tengamos bien identificados cuáles son los datos críticos que tenemos que proteger, y en esto como nosotros los que desarrollamos es muy probable que no sepamos cuáles son nuestros datos necesitamos que la gente que nos ha encomendado el sistema que los utilizadores de estos datos nos hablen nos informen nos indiquen el nivel de criticidad protección o digamos si a mí me metiera en la página abajo por un ataque de denegación del servicio puede estar cuatro días sin la página no tengo mayor problema o pueden servir así me ponen alguna foto me cambié la portada de la página no hay ningún problema.

El símbolo importante en el desarrollo que estemos totalmente involucrado que estemos totalmente conscientes de cuál es la información que se va a almacenar en ese sitio

Seguridad y el ciclo de desarrollo.

Tenemos que tener en cuenta que la seguridad de un sitio web o la seguridad informática es un proceso, lo que quiere decir, no tenemos que imaginar que vamos a desarrollar un sitio le vamos a aplicar la seguridad y se termina todo así porque la tecnología va cambiando la tecnología va mejorando la tecnología se va actualizando, entonces muchas van apareciendo muchas vulnerabilidades diariamente hoy en día hay activas más de 20 millones de vulnerabilidades correspondientes a diferentes tecnologías para el desarrollo web ya sea para el web proxy para los sistemas operativos que está alojado, para los administradores para los gestores de administración para los gestores de contenidos.

La seguridad de las aplicaciones es un proceso que no termina con la simple implementación de una clave segura sino que quiere enriqueciéndolo de manera permanente en la medida que los riesgos van avanzando, es que las amenazas se van siendo más y más importantes.



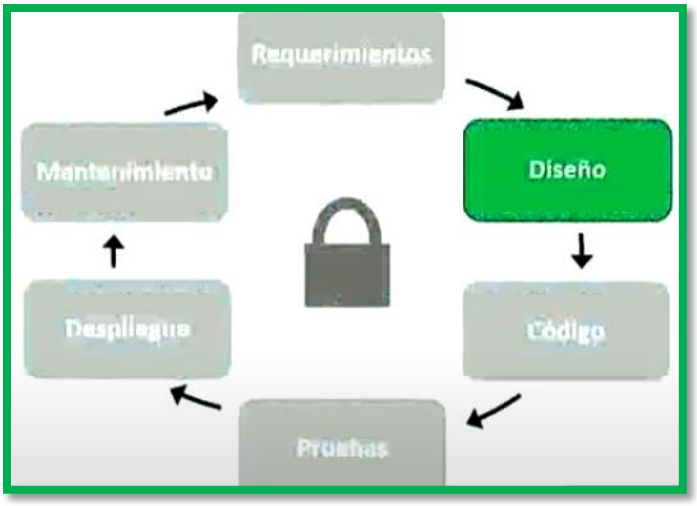
Ciclo de desarrollo. Programa Punto Digital. (2018a, julio 3).

Requerimientos.

Todo empieza por el requerimiento, lo que es el requerimiento de la aplicación tenemos que tener en cuenta cuáles son los datos críticos que tenemos que proteger y dentro de esto que tenemos que proteger son datos personales de terceros que están alcanzados por la ley de protección de datos personales.

Es fundamental que en lo que es la etapa inicial del análisis de requerimiento antes de desarrollar el sistema, estemos al tanto de los datos que vamos a alojar para saber cómo lo tenemos que proteger, por qué determinados datos la protección de determinados datos está alcanzada por la ley y el sitio tiene que incluir por ejemplo la posibilidad de que cada requerimiento del usuario se pueda borrar los datos que él puso en el sitio y se tiene que realizar de manera inmediata. Es fundamental que sepamos si vamos a almacenar datos y cuáles son esos datos.

Diseño.



Hay que tener en cuenta que es lo que tenemos que proteger hay algo que se llama seguridad por defecto que es la seguridad básica que se aplica al sitio y es darle acceso a las carpetas donde está el ojo del sitio únicamente el acceso mínimo de lectura a los usuarios que sabemos que tienen que acceder por razones estrictamente de uso.

Debemos tener en cuenta el alta de los usuarios y las claves de acceso

El usuario es el eslabón más débil de la cadena es el que se va a tratar de atacar al que se va a tratar de engañar a través de información que puede haber en el sitio.

En el diseño hay que hacer totalmente no hay que poner comentarios en todo lo que tiene que ver con el diseño de la página.

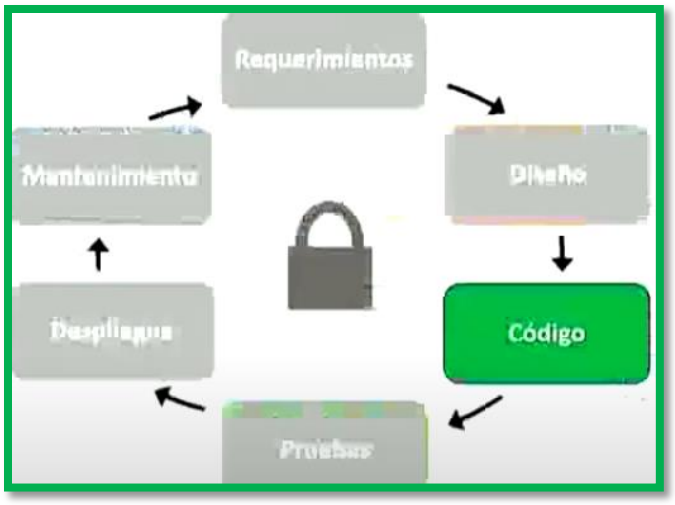
En el desarrollo del software tienen que tener en cuenta está seguridad por defecto que tiene que ver con el armado de las contraseñas seguras, con el código contacto sin demasiados comentarios ni demasiado referencias, a cómo funciona el sitio tratando de poner la menor cantidad posible de enlaces a sitios externos de los cuales no tienen el control.

Lo que hay que tratar de hacer siempre, es los request, los pedidos del usuario hacia la aplicación tienen que estar filtrados siempre por una primera frontera por un firewall idealmente por una IPs.

Es importante desarrollar que los desarrollos se centren en un servidor que atienda al peligro del tercero y sobre todo en un proxy un servidor web fuerte.

La seguridad del servidor web es fundamental y en general no suele ser parte de lo que realiza el programador entonces es fundamental comunicarse con quienes generan la plataforma de diseño de desarrollo para asegurarse que el servidor web está seguridad o que tienen implementados certificados que entran por una conexión de http segura https que tiene un servidor seguro.

Código.



En la codificación es fundamental que si se está utilizando herramientas gratuitas debemos estar seguros que tienen aplicado todos los parches, todas las actualizaciones, todas las mejoras que recomiendan los fabricantes, tienen que estar seguros que las herramientas de codificación no tienen vulnerabilidades que no están resueltas y hay que estar atentos a que no se aparezcan vulnerabilidades del día cero que son más complejas de proteger, porque en general no existe el mecanismo de defensa rápido, hay que estar atento y bajar las partes de manera inmediata y en este sentido también nos encontramos muchas veces con lenguajes de programación, la plataforma de desarrollo que una vez implementado nos seguro en actualizar a lo largo de los años.

En la seguridad es fundamental la mantención de todos los ficheros de seguridad de la aplicación de desarrollo el control de las versiones para estar seguros que aplicamos la última versión y en esto en el desarrollo hay que ser cuidadoso porque en general la aplicación de nuevas revisiones de seguridad de las herramientas de desarrollo implica quizás algún funcionamiento anómalo de la aplicación.

En general con los desarrolladores de la aplicación donde les vayan informando los bucks que van apareciendo para que en un mecanismo de mejora continua lo puedan aplicar las

nuevas versiones de desarrollo de tests probar la funcionalidad de la aplicación y pasarlo de producción lo antes posible

Entonces a la autorización de los certificados hay que estar atento en el desarrollo de la aplicación, a los ataques de cross-site scripting que requieren que la aplicación en la aplicación se haga al escape de los caracteres utilizados para hacer ataques de inyección de código o de denegación de servicio, hay que estar atento en mantenimiento a escapar estos caracteres y que no forman parte del código, siempre más recomendable que todo el código se encuentra en un servidor separado y que ha pedido el contexto del usuario se basa atrás siempre desde una base de datos y se basa presentando en contraste los usuarios.

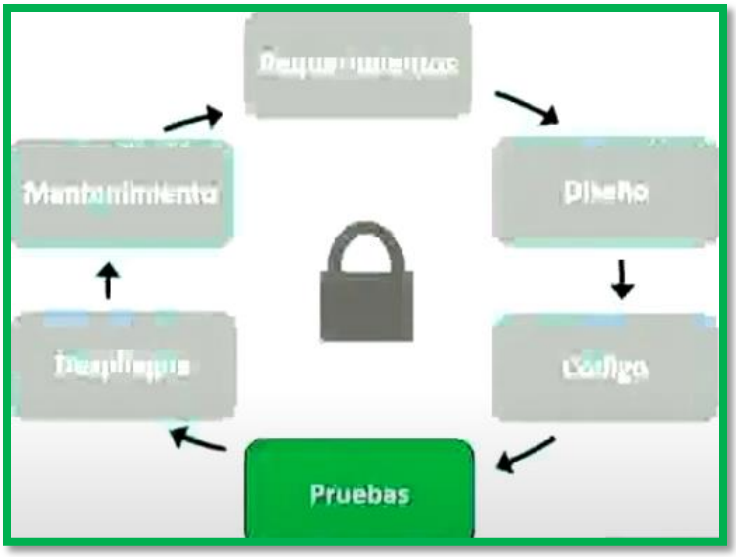
El código tiene que estar escrito de modo tal que sea:

- Fácil de mantener
- Fácil de controlar
- Fácil de aplicar los dos partes y poder ver el impacto

Hay que estar atento que todos los componentes se encuentran segurizados, hay que estar atento también a el código escrito ya que muchas veces hemos visto páginas comprometidas con código malicioso instalado y ese código para acelerar el desarrollo se

va copiando de servidor en servidor, entonces un programador quizás recibe un desarrollo ya hecho que tiene una puerta trasera y no se toma el trabajo de verificarlo.

Pruebas.



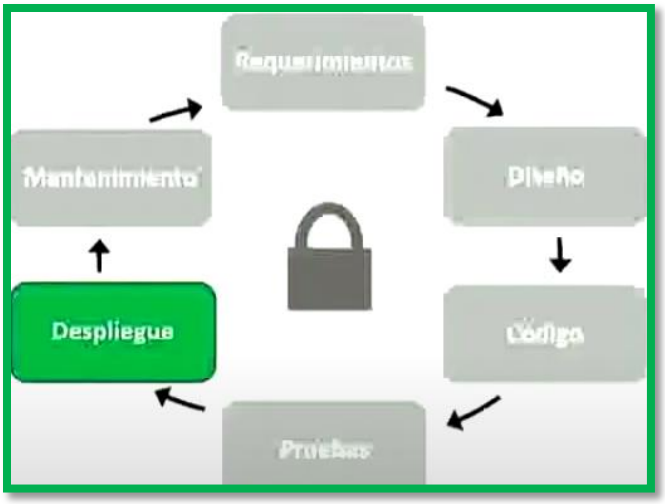
En la etapa de pruebas es convenientes que algún otro programador, ya sea un programador que desarrolló la aplicación, de una revisada al código por lo menos en los modelos más críticos y que se corre algún test, algún scan de vulnerabilidades y con estas herramientas pueden tener una primera impresión de que tan seguro es el código y en base a lo que el resultado de esta herramienta pueden profundizar en corregir aquellas vulnerabilidades que se hicieron presente.

Es muy importante que exista una etapa de prueba en un entorno distinto al entorno de producción donde se verifique la seguridad por medio de una herramienta gratuita o si es posible una de paga mejor.

Si se desea avanzar un poco mas en la seguridad, se puede realizar un penetration test que es una prueba de seguridad mucho más exhaustiva donde se imitan los mismos ataques que hacen los hackers de manera controlada para verificar que no es posible acceder desde el exterior sin contar con las autorizaciones correspondientes.

El scan de vulnerabilidades es un test que es recomendable hacer es una etapa antes de ponerlo en producción y en un entorno de test y cuando ya se corrigieron las primeras vulnerabilidades que se hicieron evidentes con este scan, avanzar entonces con un penetration test que se hace desde el exterior del lugar donde se esté desarrollando por gente que no conoce el sistema y que actúa de la misma manera que si fuera un hacker.

Despliegue.



La bajada de la aplicación ya suponemos verificada por los pagarés hecho el scan de poner habilidades hechos de tan estresantes hay que poner la aplicación en producción.

Hacemos el scan de vulnerabilidades antes de que se ponga en producción, una vez que lo hicimos y estas vulnerabilidades son corregidas, volvemos a hacer un control más en mayor profundidad y cuando determinamos que están convenientemente protegidas le damos el ok para que lo pongan en producción para que hagan despliegue.

Es fundamental en el que al momento que se hagan despliegue los equipos donde van a estar alojados tengan el sistema operativo parcheado que tengan la última versión que es la más segura que el web proxy se encuentra también actualizado en cuanto a sus partes de seguridad.

Mantenimiento:

Nos referimos en particular al mantenimiento de la seguridad, parches a la aplicación de últimas actualizaciones del sistema operativo donde está alojado el sitio

Muchas veces los firewall, los IPs, los mecanismos de protección del borde tienen también las actualizaciones periódicas, hay que estar atentos más sobre la persona que desarrolla la aplicación, la gente que nos da la plataforma tecnológica, tenemos que estar seguros que hizo las actualizaciones correspondientes, tenemos que estar seguro que la aplicación tiene un backup diario porque es una media seguridad muy importante porque ante cualquier problema y antes de ponernos inclusive a investigar la fuente del origen del incidente, una vez que el sitio está caído, una vez que el sitio fue alterado, lo primero que vamos a precisar para seguir tanto al servicio es el backup.

Etapa final del desarrollo de la aplicación.

La disposición segura, sobre todo de los datos que contienen si la aplicación tiene datos de personas, en tanto destilación, datos alcanzado por la ley de protección de datos personales, tenemos que estar seguros que esos datos están resguardados en un servidor o en un lugar que no es accesible por terceros y que nadie va a poder acceder y utilizarlos con beneficios ilícitos.

Metodologías de desarrollo seguro (Secure-SDLC)

Ciclo de vida del Desarrollo de Software (SDLC).

- Planeación y entendimiento del requerimiento:
  - Lo primero que deberíamos hacer es tratar de entender el requerimiento del cliente este cliente puede ser interno o externo obviamente y a partir de aquí habrá un analista o un arquitecto de aplicaciones que va a intentar empezar a planear el desarrollo de este este software
- Definición de métodos y técnicas para el desarrollo:
  - El arquitecto debería empezar a planear la definición los métodos las técnicas de desarrollo que se van a utilizar, empieza a contar los distintos tipos de diagramas que podemos que esté conocer, se describe gráficamente cómo va a ser la arquitectura de este desarrollo.
- Buenas prácticas y estándares:
  - Los diagramas nos ayudan mucho, por ejemplo, los famosos diagrama DFD, de este más aquí en el tiempo y más en el presente, de los diagramas UML nos van a empezar a ayudar a diagramar esta estas aplicaciones para que sea entendible por todos los que están involucrados en el desarrollo
- Revisión del diseño y banco de pruebas:
  - Una vez que ya tenemos todo pensado empezamos el tema de la revisión y el diseño
- Seguimiento de procesos y falencias detectadas:
  - Deberíamos hacer un seguimiento de progresos y tratar de ver si se detecta algún tipo de falla o algún tipo de falencia para corregir

Ciclo de vida del Desarrollo de Software (Ciclo de mejora continua)

Las etapas del ciclo de vida del desarrollo de software son: Requerimientos/análisis, diseño, implementación. Pruebas y evolución.

Primero planifico, luego tengo toda la parte de requerimientos, análisis del diseño, implementó el software, testeo este software, esto lo podemos hacer en paralelo, incluso puedo hacer una implementación en paralelo con el testing, o como generalmente los ciclos más comunes, primero se testea y luego se implementa y obviamente esto nos va a llevar a la evolución del software en donde alguien, por ejemplo, encuentra un error, esto puede ser un usuario puede ser alguien interno o externo de la organización, encuentra un error, va a llevar a planificar una evolución, qué significa, esto encontró un error lo corrijo veo luego si esta corrección no impacta en algunas partes del sistema que ya tengo funcionando, implementó la mejora y así sucesivamente la vuelvo a testear, y así sucesivamente, nos quedamos dentro de un ciclo de mejora en donde se implementa algo, se corrige, se encuentra un error, se corrige, se vuelve a implementar, se encuentra un error, se corrige y así sucesivamente

Visión de sistemas.

- Se encuentran vulnerabilidades y se corrigen.
- Se detectan los problemas externos.
- Se toman medidas después de los incidentes.

Visión holística.

- Se analiza la seguridad en todo proceso.
- Se busca la causa del problema.
- Se consideran todas las amenazas posibles.

Gestión de Riesgos

Lo que busca es primero que todos los analistas, los programadores y todos los que están relacionados con una aplicación en particular comprendan los objetivos del negocio, qué significa esto, que yo como miembro de una organización tengo que saber a qué se dedica mi empresa

- Lo primero que necesitamos es comprender los objetivos del negocio y que todos los actores las tengan claros.
- Una vez que identificamos cuáles son estos objetivos, vamos a estar en capacidad de identificar cuáles son los riesgos que enfrenta ese negocio, estos riesgos pueden ser desde el punto de vista pura y exclusivamente del negocio puede ser una estafa, un fraude o algo parecido, o riesgos técnicos en donde estos van a estar más relacionados con errores relacionados a infraestructuras sistemas etcétera.
- Una vez que sepamos cuáles son esos riesgos, la idea es crear un ranking y priorizarlos, generalmente este tipo de rankings los simbolizamos para que sea muy sencillo en un formato tipo semáforo, qué significan, riesgos altos el nivel rojo, riesgo del nivel medio con una luz amarilla y riesgo del nivel bajo con una luz verde, esto no significa que no tengamos que arreglarlos a todos pero por lo menos nos da una idea de que hay riesgos que son más urgentes y los tengo que arreglar primero
- Se definen distintos tipos de mitigación que existe para cada riesgo, en este caso, los dispositivos sería parchear inmediatamente el sistema operativo del dispositivo, en el caso que sea una aplicación instalar un parche y así sucesivamente.
- Se realizan las correcciones, se valida para sí está todo bien y nuevamente el ciclo comienza en identificar los riesgos, priorizarlos y así sucesivamente.

Metodologías de desarrollo seguro (Secure-SDLC)

Modelado de amenazas

Es una técnica formal de ahí justamente que éste sea orientado al análisis de riesgo tiene que ser totalmente formal estructurada y por otro lado tienen que ser repetible qué significa esto que si yo lo realizo hoy y lo realizas tú mañana y lo realiza otra persona pasado mañana tienen que darse las mismas condiciones para que se pueda determinar cuáles son los riesgos y ponderar estos riesgos.

El modelado de amenazas nos va a permitir determinar este ranking, utilizando estas herramientas vamos a poder saber a qué nivel de riesgo está expuesta nuestra aplicación, identificar cuáles son los riesgos, ponderar esos riesgos y hacer un ranking.

No existe ni un solo modelado de amenazas de hecho cada vez que se implementa este cada organización suele ajustarlo a sus requerimientos.

- Todo lo que sea la serie NIST 800 guion algo, son metodologías o manuales, estándares, buenas prácticas etc., que se pueden descargar en forma gratuita
- OWASP es una organización sin fines de lucro que publica herramientas y estándares para todas las personas que estamos involucrados en la arquitectura el análisis y el desarrollo de una aplicación web justamente la w viene de web

El costo real de los errores de software

"El costo de eliminar una vulnerabilidad durante la fase de diseño es hasta 60 veces menor que hacerlo en la etapa de producción"

En la etapa de codificación por nosotros agregamos código, pero éste también agregamos este una gran cantidad de bugs cuando estamos desarrollando y lo que tenemos que intentar es que estos bugs nos lleguen en la etapa de este decodificación y por lo tanto, esa curva que se incrementaría al final en producción, de corregir bugs y costo, es altísimo en corregir un bug en la etapa de producción y la estamos haciendo bajar.

- Si empezamos a codificar sin una arquitectura adecuada obviamente va a haber errores.
- Tenemos que pensar el diseño y la arquitectura antes de empezar a codificar

Modelado de amenazas SD3

- **Secure by Design**

Piensa en mitigar las amenazas y prevenir las vulnerabilidades en la etapa de diseño, recién es un enfoque totalmente proactivo en donde se empieza a pensar que cualquier aplicación que yo vaya a desarrollar en este momento va a ser vulnerable. lo que tenemos que hacer es decir, voy a desarrollar una aplicación, esta aplicación seguramente va a tener errores y esos errores van a dar lugar a que un tercero me la pueda atacar, por lo tanto lo que voy a tratar de pensar es disminuir las amenazas, pero por lo menos voy a saber cuáles son las mitigaciones y voy a tratar de minimizar esos errores

- **Secure by Default**

Es un conjunto de buenas prácticas que una de ellas es el menor privilegio es básicamente el menor privilegio nos dice que una aplicación debería ejecutarse con el mínimo privilegio necesario para que el usuario la use.

- **Secure by Deployment**

En este caso tengo la seguridad en el momento de la implementación, es decir, cada vez que voy a llevar algo a la etapa de implementación al deployment, a producción, tengo que tener o respetar buena prácticas de implementación, herramientas de administración y sobre todo tener todo lo que tenga que ver la política de parcheo, es decir, cómo voy a actualizar una aplicación determinada

Metodologías de desarrollo seguro (Secure-SDLC)

Modelo de amenazas STRIDE

Son las siglas de distintos tipos de ataques que existen actualmente y que muchos de ellos de hecho todos se pueden mitigar con el SD3.

- **Spoofing:** Ataque a la autenticación, es decir, voy a falsificar quien soy, me voy a hacer pasar por otro.
- **Tampering:** Ataques a la integridad, es decir, voy a poder modificar información.
- **Repudiation**: Son todas las técnicas de no repudio es decir yo hice algo en el sistema y luego digo no yo no fui y no lo hice esto generalmente se puede éste mitigar con todo lo que tenga que ver con criptografía.
- **Information Disclosure:** Atacan a la confidencialidad, en donde alguien, sea por el motivo que sea y sea como lo haya hecho, ataca un servidor o n servidores o el correo electrónico o lo que haya sido y se divulgó información que en principio era confidencial.
- **Denial of Service:** Atacan la disponibilidad de un servicio determinado
- Elevation of Privilege: En donde un usuario normal, sin ningún privilegio dentro de la organización, logró ejecutar algo con permisos de sistema, con permisos administrativos, con permisos de root, etcétera, de forma tal que va a lograr más acceso a lo que realmente debería tener dentro de la compañía o dentro de la organización

Modelo de amenazas DREAD

Nos va a permitir determinar en base, una fórmula muy sencilla, cuál es el daño potencial que puede tener una aplicación, es decir si alguien ataca mi aplicación, qué daño puede sufrir la aplicación, el sistema, la infraestructura y sus usuarios, el nivel de reproductibilidad que puede tener el ataque y el nivel de explotabilidad que puede tener el ataque, los usuarios afectados también puede ser alta y el nivel de descubrimiento que tiene la vulnerabilidad y también me va a indicar qué tan ríen qué tan nivel de riesgo esto.

- Damage potencial.
- Reproducibility.
- Explotability.
- Affected User.
- Discoverability.

$$RISK = \frac{D+R+E+A+D}{5}$$

Secure - SDLC

A todo el ciclo de vida del desarrollo de software que estamos incluyendo principios de seguridad buenas prácticas y estándares reconocidos del mercado.

Estamos agregando todos los principios de seguridad al ciclo de vida del desarrollo de software.

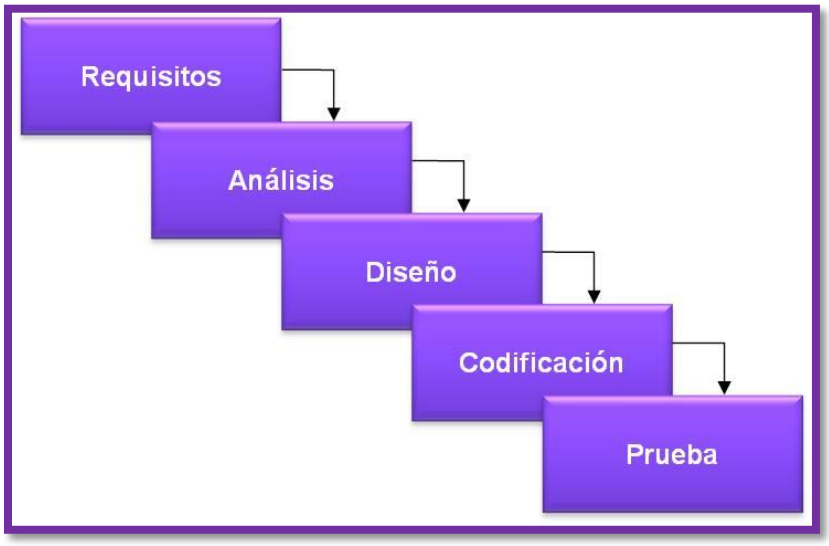
Tenemos el ciclo de vida de desarrollo de software, le agregamos seguridad para construir lo que se conoce como el ciclo de vida de desarrollo seguro.

Tú puedes hacer las aplicaciones o los desarrollos como quieras, en el lenguaje que quieras, en la infraestructura que quieras, aplicando la arquitectura que quieras siempre buscando cubrir los objetivos del negocio, tener en cuenta siempre como base ese objetivo del negocio y como siempre cumpliendo que su misión sea la funcionalidad, pero esta vez implementando la seguridad en la funcionalidad, que no sólo funcione sino que además sea seguro que además cumpla con todos los parámetros las buenas práctica, aplicados directamente en el desarrollo y en los procesos que nos llevan a tener unas aplicaciones y cumplir los objetivos de manera tanto funcional como segura.

Metodologías de desarrollo seguro (SDLC).

Ciclo de vida del Desarrollo de Software (SDLC).

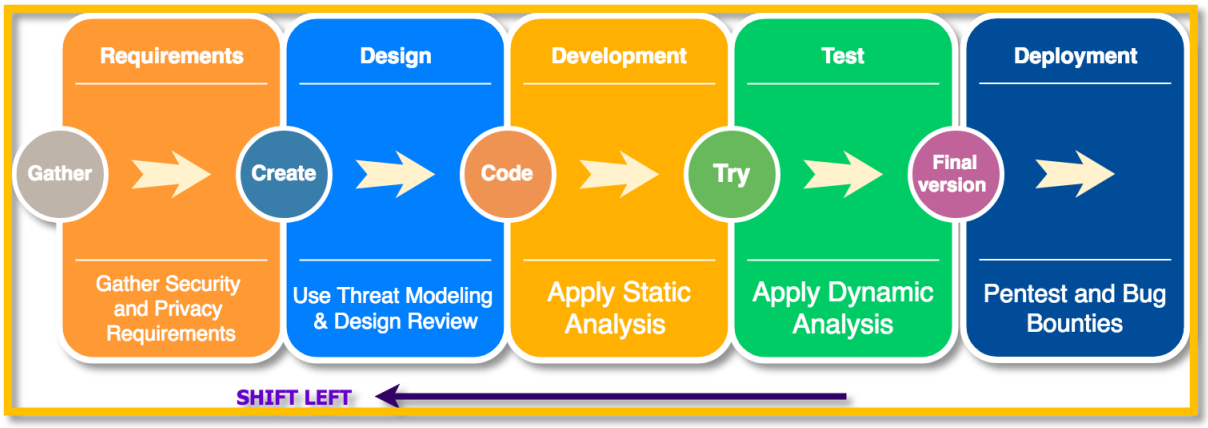
Es una secuencia estructurada que cuenta con las siguientes etapas:



- Planeación y entendimiento del requerimiento:
    - Lo primero que deberíamos hacer es tratar de entender el requerimiento del cliente este cliente puede ser interno o externo obviamente y a partir de aquí habrá un analista o un arquitecto de aplicaciones que va a intentar empezar a planear el desarrollo de este este software
- Definición de métodos y técnicas para el desarrollo:
    - El arquitecto debería empezar a planear la definición los métodos las técnicas de desarrollo que se van a utilizar, empieza a contar los distintos tipos de diagramas que podemos que esté conocer, se describe gráficamente cómo va a ser la arquitectura de este desarrollo.
- Buenas prácticas y estándares:
    - Los diagramas nos ayudan mucho, por ejemplo, los famosos diagrama DFD, de este más aquí en el tiempo y más en el presen te, de los diagramas UML

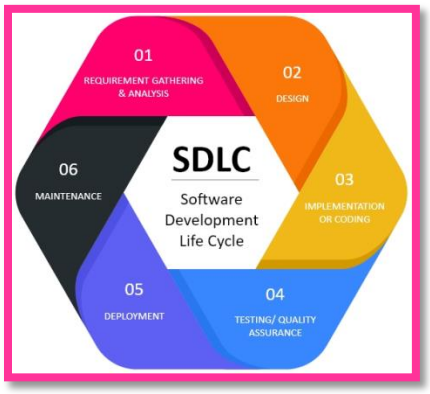
nos van a empezar a ayudar a diagramar esta estas aplicaciones para que sea entendible por todos los que están involucrados en el desarrollo.

- Revisión del diseño y banco de pruebas:
    - Una vez que ya tenemos todo pensado empezamos el tema de la revisión y el diseño.
- Seguimiento de procesos y falencias detectadas:
    - Deberíamos hacer un seguimiento de progresos y tratar de ver si se detecta algún tipo de falla o algún tipo de falencia para corregir.



Ciclo de vida del Desarrollo de Software (Ciclo de mejora continua).

Las etapas del ciclo de vida del desarrollo de software son: Requerimientos/análisis, diseño, implementación, Pruebas y evolución.



Primero planifico, luego tengo toda la parte de requerimientos, análisis del diseño, implementó el software, testeo este software, esto lo podemos hacer en paralelo, incluso puedo hacer una implementación en paralelo con el testing, o como generalmente los ciclos más comunes, primero se testea y luego se implementa y obviamente esto nos va a llevar a la evolución del software en donde alguien, por ejemplo, encuentra un error, esto puede ser un usuario puede ser alguien interno o externo de la organización, encuentra un error, va a llevar a planificar una evolución, qué significa, esto encontró un error lo corrijo veo luego si esta corrección no impacta en algunas partes del sistema que ya tengo funcionando, implementó la mejora y así sucesivamente la vuelvo a testear, y así sucesivamente, nos quedamos dentro de un ciclo de mejora en donde se implementa algo, se corrige, se encuentra un error, se corrige, se vuelve a implementar, se encuentra un error, se corrige y así sucesivamente.

- Visión de sistemas
    - Se encuentran vulnerabilidades y se corrigen.
    - Se detectan los problemas externos.
    - Se toman medidas después de los incidentes.

- Visión holística
    - Se analiza la seguridad en todo proceso.
    - Se busca la causa del problema.
    - Se consideran todas las amenazas posibles.

Gestión de Riesgos.

Lo que busca es primero que todos los analistas, los programadores y todos los que están relacionados con una aplicación en particular comprendan los objetivos del negocio, qué significa esto, que yo como miembro de una organización tengo que saber a qué se dedica mi empresa.

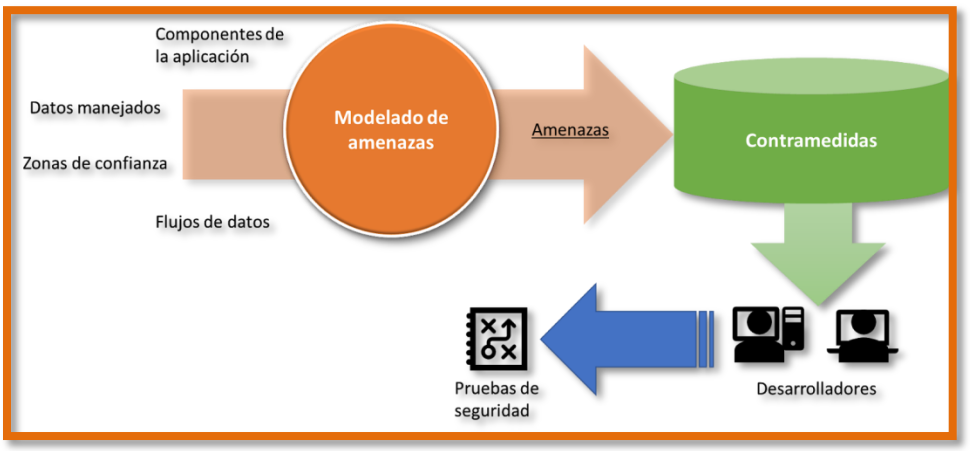
- Lo primero que necesitamos es comprender los objetivos del negocio y que todos los actores las tengan claros.
- Una vez que identificamos cuáles son estos objetivos, vamos a estar en capacidad de identificar cuáles son los riesgos que enfrenta ese negocio, estos riesgos pueden ser desde el punto de vista pura y exclusivamente del negocio puede ser una estafa, un fraude o algo parecido, o riesgos técnicos en donde estos van a estar más relacionados con errores relacionados a infraestructuras sistemas etcétera.
- Una vez que sepamos cuáles son esos riesgos, la idea es crear un ranking y priorizarlos, generalmente este tipo de rankings los simbolizamos para que sea muy sencillo en un formato tipo semáforo, qué significan, riesgos altos el nivel rojo, riesgo del nivel medio con una luz amarilla y riesgo del nivel bajo con una luz verde, esto no significa que no tengamos que arreglarlos a todos pero por lo menos

nos da una idea de que hay riesgos que son más urgentes y los tengo que arreglar primero.

- Se definen distintos tipos de mitigación que existe para cada riesgo, en este caso, los dispositivos sería parchear inmediatamente el sistema operativo del dispositivo, en el caso que sea una aplicación instalar un parche y así sucesivamente.
- Se realizan las correcciones, se valida para sí está todo bien y nuevamente el ciclo comienza en identificar los riesgos, priorizarlos y así sucesivamente.

Modelado de amenazas.

Es una técnica formal de ahí justamente que éste sea orientado al análisis de riesgo tiene que ser totalmente formal estructurada y por otro lado tienen que ser repetible qué significa esto que si yo lo realizo hoy y lo realizas tú mañana y lo realiza otra persona pasado mañana tienen que darse las mismas condiciones para que se pueda determinar cuáles son los riesgos y ponderar estos riesgos.



El modelado de amenazas nos va a permitir determinar este ranking, utilizando estas herramientas vamos a poder saber a qué nivel de riesgo está expuesta nuestra aplicación, identificar cuáles son los riesgos, ponderar esos riesgos y hacer un ranking.

No existe ni un solo modelado de amenazas de hecho cada vez que se implementa este cada organización suele ajustarlo a sus requerimientos.

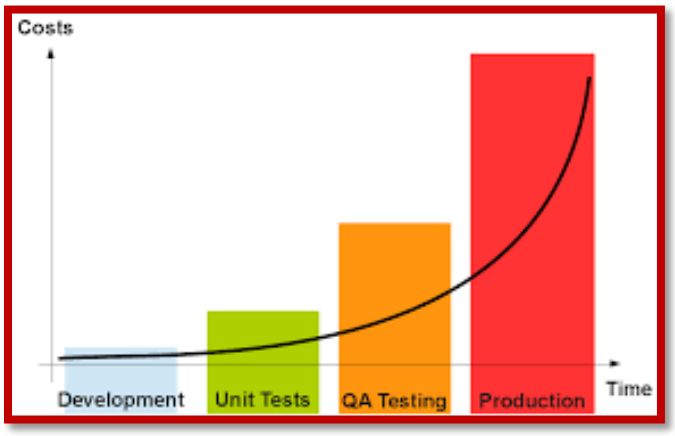
- Todo lo que sea la serie NIST 800 guion algo, son metodologías o manuales, estándares, buenas prácticas etc., que se pueden descargar en forma gratuita.

- OWASP es una organización sin fines de lucro que publica herramientas y estándares para todas las personas que estamos involucrados en la arquitectura el análisis y el desarrollo de una aplicación web justamente la w viene de web

“El costo de eliminar una vulnerabilidad durante la fase de diseño es hasta 60 veces menor que hacerlo en la etapa de producción”.

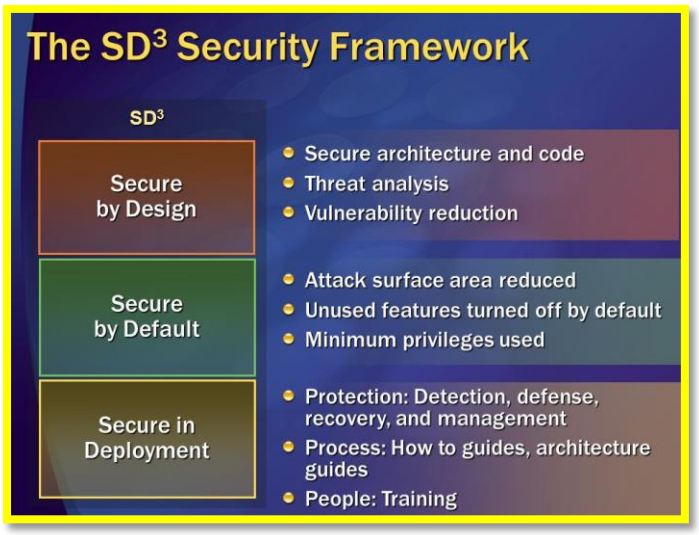
El costo real de los errores de software.

En la etapa de codificación por nosotros agregamos código, pero éste también agregamos este una gran cantidad de bugs cuando estamos desarrollando y lo que tenemos que intentar es que estos bugs nos lleguen en la etapa de este decodificación y por lo tanto, esa curva que se incrementaría al final en producción, de corregir bugs y costo, es altísimo en corregir un bug en la etapa de producción y la estamos haciendo bajar.



- Si empezamos a codificar sin una arquitectura adecuada obviamente va a haber errores.
- Tenemos que pensar el diseño y la arquitectura antes de empezar a codificar

Modelado de amenazas SD3.



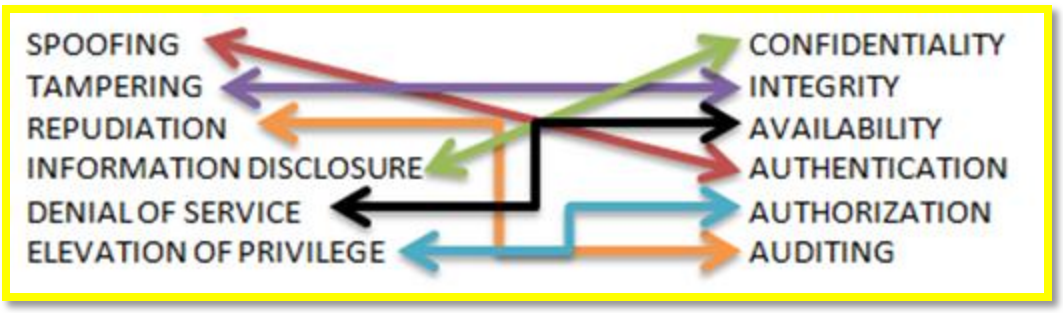
- **Secure by Design:** Piensa en mitigar las amenazas y prevenir las vulnerabilidades en la etapa de diseño, recién es un enfoque totalmente proactivo en donde se empieza a pensar que cualquier aplicación que yo vaya a desarrollar en este momento va a ser vulnerable. lo que tenemos que hacer es decir, voy a desarrollar una aplicación, esta aplicación seguramente va a tener errores y esos errores van a dar lugar a que un tercero me la pueda atacar, por lo tanto lo que voy a tratar de pensar es disminuir las amenazas, pero por lo menos voy a saber cuáles son las mitigaciones y voy a tratar de minimizar esos errores.
- **Secure by Default:** Es un conjunto de buenas prácticas que una de ellas es el menor privilegio es básicamente el menor privilegio nos dice que una aplicación debería ejecutarse con el mínimo privilegio necesario para que el usuario la use.
- **Secure by Deployment:** En este caso tengo la seguridad en el momento de la implementación, es decir, cada vez que voy a llevar algo a la etapa de implementación al deployment, a producción, tengo que tener o respetar buena prácticas de implementación, herramientas de administración y sobre todo tener todo lo que tenga que ver la política de parcheo, es decir, cómo voy a actualizar una aplicación determinada.

Modelo de amenazas STRIDE.

Son las siglas de distintos tipos de ataques que existen actualmente y que muchos de ellos de hecho todos se pueden mitigar con el SD3.

- Spoofing: Ataque a la autenticación, es decir, voy a falsificar quien soy, me voy a hacer pasar por otro.
- Tampering: Ataques a la integridad, es decir, voy a poder modificar información.
- Repudiation: Son todas las técnicas de no repudio es decir yo hice algo en el sistema y luego digo no yo no fui y no lo hice esto generalmente se puede éste mitigar con todo lo que tenga que ver con criptografía.
- Information Disclosure: Atacan a la confidencialidad, en donde alguien, sea por el motivo que sea y sea como lo haya hecho, ataca un servidor o n servidores o el correo electrónico o lo que haya sido y se divulgó información que en principio era confidencial.
- Denial of Service: Atacan la disponibilidad de un servicio determinado
- Elevation of Privilege: En donde un usuario normal, sin ningún privilegio dentro de la organización, logró ejecutar algo con permisos de sistema, con permisos administrativos, con permisos de root, etcétera, de forma tal que va a lograr más

acceso a lo que realmente debería tener dentro de la compañía o dentro de la organización



Modelo de amenazas DREAD

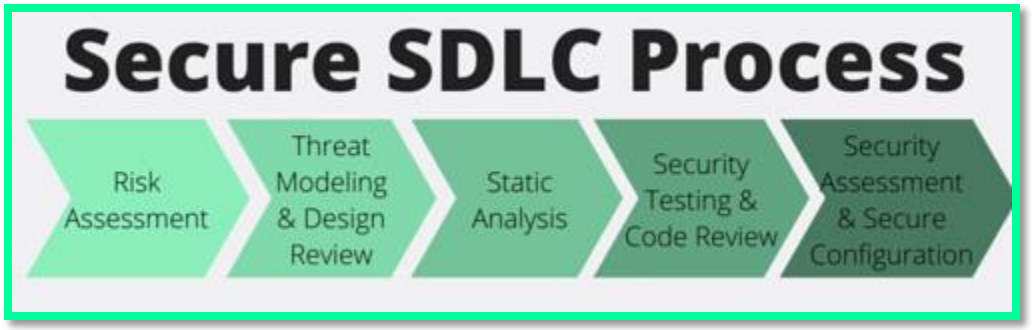
Nos va a permitir determinar en base, una fórmula muy sencilla, cuál es el daño potencial que puede tener una aplicación, es decir si alguien ataca mi aplicación, qué daño puede sufrir la aplicación, el sistema, la infraestructura y sus usuarios, el nivel de reproductibilidad que puede tener el ataque y el nivel de explotabilidad que puede tener el ataque, los usuarios afectados también puede ser alta y el nivel de descubrimiento que tiene la vulnerabilidad y también me va a indicar qué tan ríen qué tan nivel de riesgo esto.

Secure – SDLC.

A todo el ciclo de vida del desarrollo de software que estamos incluyendo principios de seguridad buenas prácticas y estándares reconocidos del mercado.

Estamos agregando todos los principios de seguridad al ciclo de vida del desarrollo de software.

Tenemos el ciclo de vida de desarrollo de software, le agregamos seguridad para construir lo que se conoce como el ciclo de vida de desarrollo seguro.



Tú puedes hacer las aplicaciones o los desarrollos como quieras, en el lenguaje que quieras, en la infraestructura que quieras, aplicando la arquitectura que quieras siempre buscando cubrir los objetivos del negocio, tener en cuenta siempre como base ese objetivo del negocio y como siempre cumpliendo que su misión sea la funcionalidad, pero esta vez implementando la seguridad en la funcionalidad, que no sólo funcione sino que además sea seguro que además cumpla con todos los parámetros las buenas práctica, aplicados directamente en el desarrollo y en los procesos que nos llevan a tener unas aplicaciones y cumplir los objetivos de manera tanto funcional como segura.

Desarrollo seguro

Referencia.

- Programa Punto Digital. (2018, 3 agosto). *Desarrollo seguro de aplicaciones web* [Vídeo]. YouTube. https://www.youtube.com/watch?v=xyw_5Rd-sf0

- ElevenPaths. (2016, 26 agosto). *#11PathsTalks: Metodologías de desarrollo seguro (Secure-SDLC)* [Vídeo]. YouTube. https://www.youtube.com/watch?v=eMs2fErek_c&t=2s

- Ember Dev. (2019, 25 junio). *¿Qué es el desarrollo seguro?* [Vídeo]. YouTube. https://www.youtube.com/watch?v=qZodhPKZZYI&t=1s

- AERTIC. (2020, 11 agosto). *Desarrollo seguro y seguridad en el ciclo de vida del software- SSHTEAM* [Vídeo]. YouTube. https://www.youtube.com/watch?v=GgcOB2Etlf4&t=1s