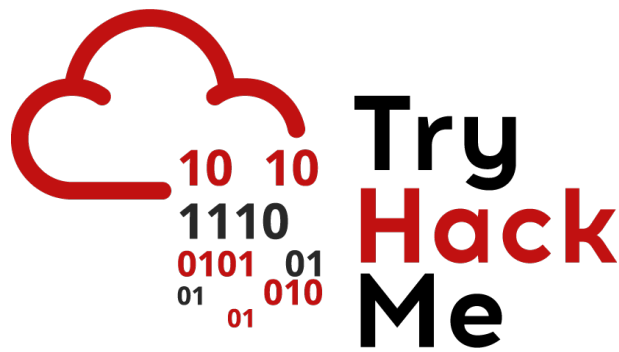


Writeup: Sala *SDLC*

Autor: Ismaeldevs

Plataforma: TryHackMe

4 de julio de 2025



Índice

1. Introducción	2
2. Sala	2
2.1. Tarea 1 – Introducción	2
2.2. Tarea 2 - SDLC	2
2.3. Tarea 3 - Fases del SDLC Parte 1	3
2.4. Tarea 4 - Fases del SDLC Parte 2	3
2.5. Tarea 5 - Keep CALMS	4
2.6. Tarea 6 - Métricas de DevOps	4
2.7. Tarea 7 - Producción de los droides	5
3. Conclusión sobre la Sala	7

1. Introducción

En la sala **SDLC (Software Development Life Cycle)** aprenderemos los fundamentos del ciclo de vida del desarrollo de software, sus fases clave, el rol de la seguridad en cada etapa y cómo metodologías como CALMS permiten una integración continua y eficiente.

2. Sala

2.1. Tarea 1 – Introducción

En esta primera tarea nos proporciona una breve introducción y sienta los objetivos de aprendizaje de la sala.

Pregunta: I'm ready to start!

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

2.2. Tarea 2 - SDLC

Aprendemos en profundidad qué es el **SDLC**, el cual es un marco de trabajo que guía el desarrollo de software a través de una serie de fases organizadas que suelen dividirse en 6-8 fases.

- **Planificación**
- **Definición de requisitos**
- **Diseño y creación de prototipos**
- **Desarrollo de software**
- **Pruebas**
- **Despliegue**
- **Operaciones y mantenimiento**

También explica cómo al aplicar SDLC, es posible estructurar proyectos, reducir errores, mejorar la colaboración entre equipos y garantizar entregas consistentes y confiables.

Una vez que entendemos qué es y como funciona el SDLC, procedemos a responder las siguientes preguntas.

Pregunta: How many phases can an SDLC have? (Format X-Y)

Respuesta: 6-8

2.3. Tarea 3 - Fases del SDLC Parte 1

En esta tarea describe como las primeras fases se centran en desglosar el proyecto o la aplicación antes de pasar al desarrollo de la aplicación. Estas fases son:

1. **Planning**
2. **Requirements Definition**
3. **Design and Prototyping**
4. **Software Development**

Esto es con el fin de estudiar en detalle las necesidades funcionales y no funcionales de un software y cómo estas etapas sientan las bases para un desarrollo eficiente y alineado con las expectativas del cliente o usuario final.

Después que comprendemos la primera parte de la fases del SDLC, pasamos a responder las siguientes preguntas.

Pregunta: What phase focuses on determining the first idea for a prototype?

Respuesta: Requirements Definition

Pregunta: What stage is also known as the Feasibility Stage?

Respuesta: Planning Stage

Pregunta: When do you outline the user interfaces and network requirements?

Respuesta: Design and Prototyping

2.4. Tarea 4 - Fases del SDLC Parte 2

Ahora conoceremos las fases finales que se centran en probar la aplicación antes del lanzamiento y en mantener su correcto funcionamiento posteriormente. Estas fases son:

1. **Pruebas**
2. **Implementación**
3. **Operaciones y mantenimiento**

Una vez entendido las fases, posterior a eso, vamos a responder las siguientes preguntas.

Pregunta: What phase focuses on handling issues or bugs reported by end-users?

Respuesta: **Operations and Maintenance**

Pregunta: What phase involves releasing new versions of software?

Respuesta: **Deployment**

Pregunta: What phase ensures software meets the standards defined in the requirements phase?

Respuesta: **Testing**

2.5. Tarea 5 - Keep CALMS

Aprenderemos sobre el acrónimo **CALMS**, un modelo para evaluar la madurez de equipos DevOps del cual consiste en:

- **Cultura**
- **Automatización**
- **Infalible**
- **Medición**
- **Compartir**

Nos explica cómo estos cinco pilares permiten integrar seguridad, colaboración y eficiencia, y refuerzan la necesidad de que los equipos trabajen con procesos iterativos, datos objetivos y aprendizaje compartido.

Una vez comprendido el modelo CALMS, pasamos a responder la siguiente pregunta en base lo aprendido.

Pregunta: What does CALMS stand for?

Respuesta: **Culture, Automation, Lean, Measurement, Sharing**

2.6. Tarea 6 - Métricas de DevOps

Conoceremos las métricas más relevantes en entornos DevOps, orientadas a medir eficiencia, calidad y seguridad. Estas se tratan de indicadores como frecuencia de despliegue, tiempo de recuperación ante fallos, tasa de cambios exitosos y demás

criterios que ayudan a tomar decisiones fundamentadas, mejorar operaciones y garantizar estabilidad y agilidad.

Posterior a entender las Métricas, procederemos a responder las siguientes preguntas.

Pregunta: What 2 metrics are used to measure deployment agility?

Respuesta: **Deployment speed and frequency**

Pregunta: What is an essential rate for engineers in Production environments to know if code meets security requirements?

Respuesta: **Failure rate**

Pregunta: What is the measurement for recovery time after a failure?

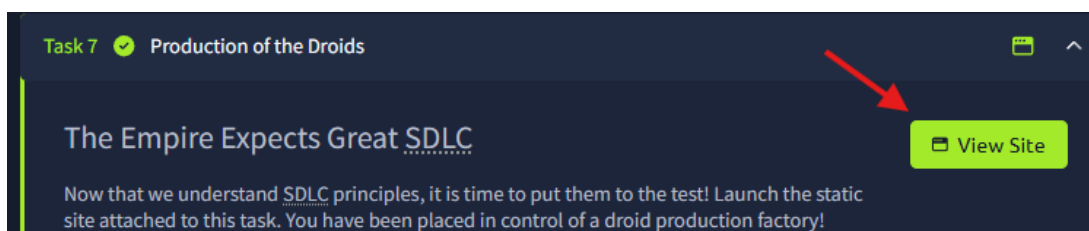
Respuesta: **MTTR**

2.7. Tarea 7 - Producción de los droides

Ahora que comprendemos los principios del ciclo de vida del desarrollo de software (SDLC), procederemos a poner en práctica los conocimientos mediante un ejercicio donde nos asignan el control de una fábrica de producción de droides.

Pregunta: What is the flag that you receive once you have doubled the empire's investment?

Una vez que comprendemos la consigna de la tarea, desplegaremos el sitio haciendo clic en **View Site** en el lado superior.



Ahora debemos decidir el número de desarrolladores y sprints. Una vez decidido, haremos clic en **Next**.

SDLC Factory Game

Select amount of sprints and developers.

Total Budget:

Cost of developer per sprint:

Number of developers:

Number of sprints:

Budget Remaining:

\$1,000,000

\$3,000

\$340,000

Next

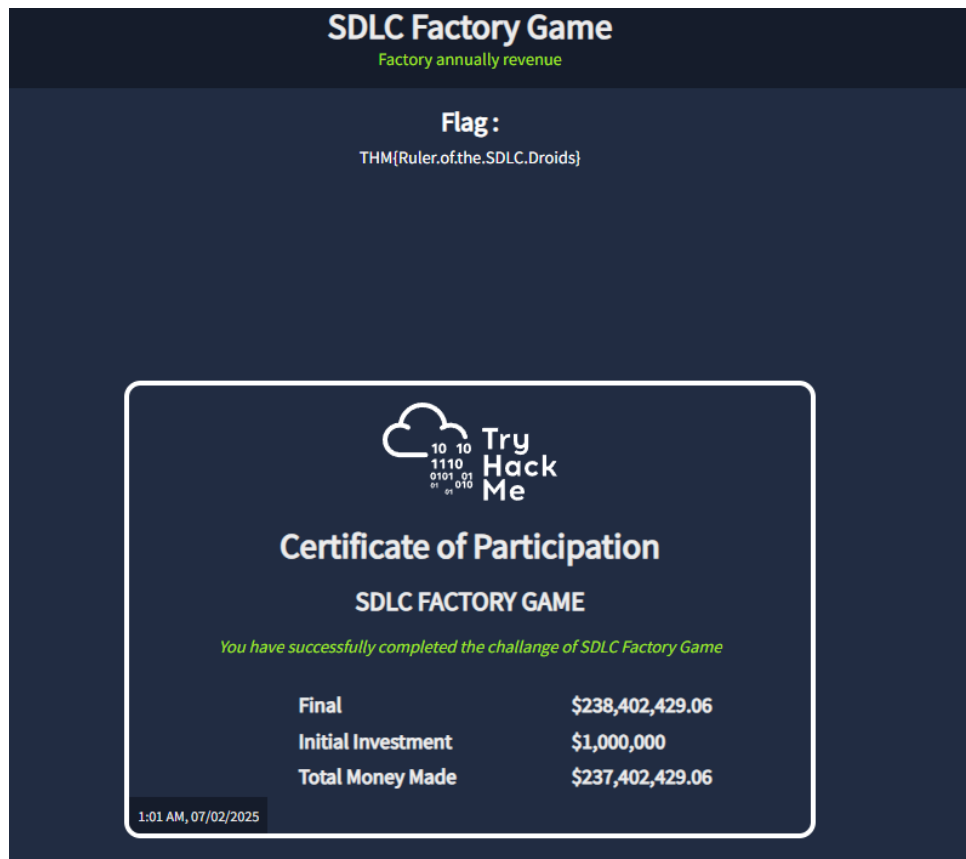
Es momento de asignar estos sprints a las distintas fases del ciclo de vida del desarrollo de software (SDLC). A medida que se añada sprints, hay que prestar mucha atención a cómo esto afectará la eficacia de la fábrica.

< Home
Remaining Sprints : 0

	Planning	<input type="button" value="-10"/> <input type="button" value="-5"/> <input type="button" value="-1"/> <input style="border: 2px solid red;" type="button" value="20"/> <input type="button" value="+1"/> <input type="button" value="+5"/> <input type="button" value="+10"/>	Game Metrics Rate of new bugs Cost to fix bugs Droid production rate Percentage of defective droids Droid production cost Droid upsell percentage Droid sell cost	Lower Limit 0% \$2933 100 0% \$10000 20% \$12000	Upper Limit 5% \$1000 618 5% \$25644 412% \$131300
	Define Requirements	<input type="button" value="-10"/> <input type="button" value="-5"/> <input type="button" value="-1"/> <input style="border: 2px solid red;" type="button" value="30"/> <input type="button" value="+1"/> <input type="button" value="+5"/> <input type="button" value="+10"/>			
	Design and Prototyping	<input type="button" value="-10"/> <input type="button" value="-5"/> <input type="button" value="-1"/> <input style="border: 2px solid red;" type="button" value="10"/> <input type="button" value="+1"/> <input type="button" value="+5"/> <input type="button" value="+10"/>			
	Software Development	<input type="button" value="-10"/> <input type="button" value="-5"/> <input type="button" value="-1"/> <input style="border: 2px solid red;" type="button" value="20"/> <input type="button" value="+1"/> <input type="button" value="+5"/> <input type="button" value="+10"/>			
	Testing	<input type="button" value="-10"/> <input type="button" value="-5"/> <input type="button" value="-1"/> <input style="border: 2px solid red;" type="button" value="10"/> <input type="button" value="+1"/> <input type="button" value="+5"/> <input type="button" value="+10"/>			
	Deployment	<input type="button" value="-10"/> <input type="button" value="-5"/> <input type="button" value="-1"/> <input style="border: 2px solid red;" type="button" value="10"/> <input type="button" value="+1"/> <input type="button" value="+5"/> <input type="button" value="+10"/>			
	Operations and Maintenance	<input type="button" value="-10"/> <input type="button" value="-5"/> <input type="button" value="-1"/> <input style="border: 2px solid red;" type="button" value="10"/> <input type="button" value="+1"/> <input type="button" value="+5"/> <input type="button" value="+10"/>			

Start Game

Después de encontrar la distribución óptima de sprints, procedemos en marcha la producción haciendo clic en **Start Game** e iremos haciendo clic en **Next month**. De esta manera logramos duplicar la inversión del imperio y recibiremos la flag.



Respuesta: **THM{Ruler.of.the.SDLC.Droids}**

3. Conclusión sobre la Sala

Al finalizar esta sala hemos logrado entender la importancia de estructurar el desarrollo de software mediante un ciclo de vida bien definido. Aprendimos cómo cada fase (desde la planificación hasta la entrega) cumplen un rol esencial en la calidad y seguridad del producto final. Además, obtuvimos conocimientos sobre principios de DevOps, métricas claves y buenas prácticas que permiten a los equipos colaborar de forma ágil y segura.