



Ingeniería en computación  
Universidad de La Serena  
II Semestre, 2020  
Versión número 1

# Determinación del ciclo de vida

**AIPA**

*Miércoles 11 de noviembre*

**Integrantes:**

Yair Gallardo  
Norton Irarrázabal  
Sebastian Rojas

**Docente**

Guillermo Leyton.

## Propósito

Este documento tiene como finalidad determinar el ciclo de vida para el software AIPA basado en:

- Las características del problema.
- Las características del software.
- Las características del ámbito de trabajo.
- Las características de los ciclos de vida.

De esta forma determinar el ciclo de vida acorde al mismo.

## Descripción del problema

El uso de inteligencia artificial en el ámbito de los jugadores virtuales, es una **herramienta de uso investigativo y publicitario** que cuenta con larga data dentro de los últimos 30 años. En el área investigativa, se pretende encontrar el alcance de las capacidades de simulación racional con las que cuentan las máquinas; el logro de altos niveles por estos jugadores virtuales suele generar gran interés en el público.

Otra perspectiva a valorar es el amplio mercado de los videojuegos, donde la dificultad, la búsqueda de retos y experiencias satisfactorias suelen ser una de las áreas de interés para los jugadores más exigentes.

Intereses que pueden ser abordados por estos jugadores virtuales, siendo un reto y por otra una “ayuda” para jugadores menos experimentados que busquen encontrar o descubrir nuevas jugadas que faciliten su éxito mejorando su experiencia y satisfacción. Por consiguiente es de interés para empresas desarrolladoras que buscan satisfacer a este tipo de clientes.

Si bien el software a desarrollar se desenvolverá en un ámbito concreto, juego de estrategia **ajedrez**; el enfoque es el mismo, siendo un buen prototipo aplicable en otros entornos.

## Características del problema

1. ¿El problema es lineal?  
**No.** Una jugada puede ser respondida con distintas jugadas, en las que cada estado tiene un valor diferente dependiendo de la instancia del juego (inclusive teniendo las mismas piezas).
2. ¿El problema es estático?  
**Sí.** el entorno no varía mientras se realizan los cálculos en cada turno.
3. ¿El problema es síncrono?  
**Sí (al menos el motor de juego).** Es un juego por turno, lo que delimita el avance de piezas o procesos del juego.

4. ¿El problema es determinista?  
**No (es estratégico).** Un conjunto de jugadas no determina al ganador, pues también se involucra un agente externo no totalmente previsible.
5. ¿El problema es estable?  
**No (semiestable).**
6. ¿El problema es azaroso?  
**No.** No puede predecirse un evento a través del azar.
7. ¿El problema es espontáneo?  
**No.** Es un juego de estrategia, por lo que sus eventos son mayormente planificados o estructurados para que así se den.
8. ¿El problema es uni sistémico?  
**No,** requiere al menos dos sistemas. Uno para el juego y otros para los agente involucrados)
9. ¿El problema es isomorfo?  
**No.**

## Características del software

- El software para su funcionamiento hace uso de agentes inteligentes definidos por el siguiente entorno de trabajo:
  - Totalmente observable: Acceso al estado completo del medio en cada momento.
  - Estratégico: El medio es determinista excepto para las acciones del otro agente.
  - Secuencial: Las acciones que se realizan a corto plazo pueden afectar a largo plazo.
  - Multiagente: Más de un agente en el sistema.
  - Discreto: Conjunto discreto de percepciones y acciones.
  - Estático: El entorno no cambia mientras el agente está deliberando.
- El software requiere de varias iteraciones para conseguir que se cumplan la totalidad de los requisitos, en el sentido de lograr ciertamente credibilidad en los resultados.
- Tiene dos funciones principales:
  - Que aprenda por sí solo a jugar ajedrez.
  - Que sirva como guía a un jugador de ajedrez humano.
- El núcleo del software se centra en los agentes inteligentes.

## Características del ámbito de trabajo

- Grupo de desarrollo de 3 integrantes, con un compromiso alto y una flexibilidad y disposición media. Sin experiencia práctica en el tipo de desarrollo
- Grupo SQA de dos integrantes con un compromiso, flexibilidad y disposición desconocida. Sin experiencia práctica en el tipo de desarrollo

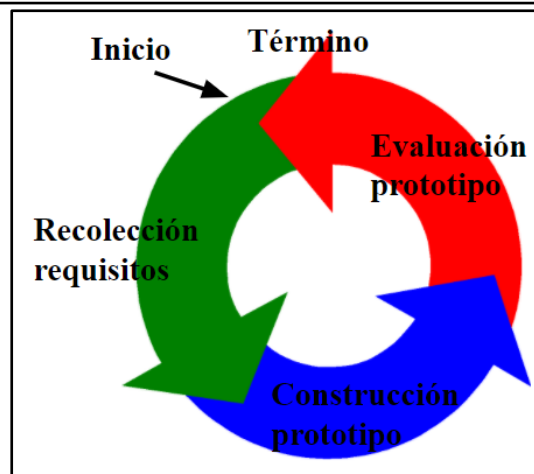
- Plazo estimado de 13 semanas de las que quedan 9 semanas.
- Se requiere velocidad en la implementación por factor tiempo.
- No se posee ni se requiere experto.
- Se desconoce el nivel de equipo necesario, pero se cree que se cuenta con los elementos necesarios para su desarrollo. En caso de que no sea así, se debe delimitar los criterios de éxito

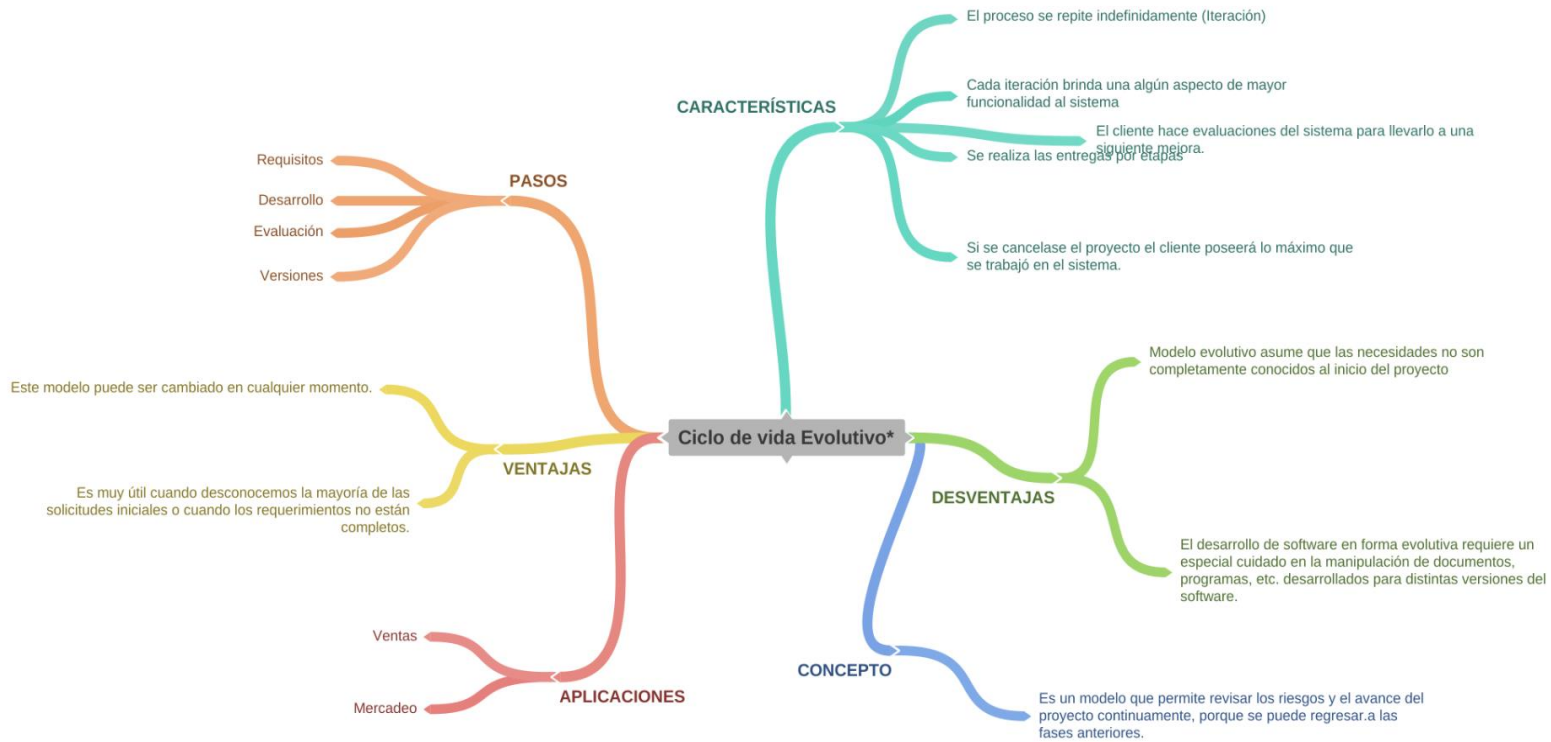
## Características del ciclo de vida seleccionado

### Ciclo de vida Evolutivo

Los evolutivos son modelos cíclicos, permiten desarrollar versiones cada vez más completas y complejas, hasta llegar al objetivo deseado.

- Los requerimientos cambian con rapidez.
- El cliente es reacio a especificar los requerimientos con mayor detalle.
- Los ciclos se repiten hasta tener un producto completo.
- Los desarrolladores no son expertos en el área de aplicación, no son jugadores profesionales de ajedrez.
- Requiere de una velocidad de desarrollo rápida.
- Se va completando el producto de software, a medida que avanza el proyecto a partir de un prototipo.
- Se comienza desarrollando el prototipo con los requerimientos más claros y completos.
- Se considera prototipo a:
  - Un programa que implemente algunas de las funciones del software.
  - Un modelo en papel.
- Incremental.
- Entrega de versiones “incompletas” pero funcionales.
- Flexible.
- La especificación puede desarrollarse de forma creciente.
- En la etapa de diseño podemos encontrar algún problema con los requerimientos y es menos dificultoso retornar a actividades de requerimientos para replantear o cambiar especificaciones.





De acuerdo a lo anterior se ha determinado que el ciclo de vida que se adecua a nuestro software es el siguiente:

- Ciclo de vida evolutivo.

#### Justificaciones:

- Existe desconocimiento en el apartado técnico, por lo que se deberá partir de requisitos genéricos que se irán detallando a medida que se investigue la solución final.
- Se cuenta con la participación activa del grupo desarrollador y docente como clientes finales, por lo que la evaluación constante de prototipos es factible.
- Se requiere un desarrollo rápido ya que se ha agotado  $\frac{1}{3}$  del tiempo estimado.
- No se está seguro de la arquitectura que tendrá el software.