



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

Especificación de requisitos

AIPA

Domingo 13 de Octubre

Integrantes:

Yair Gallardo
Norton Irarrázabal
Sebastian Rojas

Docente

Guillermo Leyton.

1. Introducción

1.1. Propósito

Este documento tiene como finalidad especificar los requisitos del software AIPA según el estándar IEEE 830, para definir ámbito del sistema, especificar restricciones y dependencias, detallar requisitos específicos (requerimientos funcionales y no funcionales).

Debe ser preciso, establecer límites y eliminar ambigüedades, ya que el software debe cumplir con aquello acordado con el cliente.

El documento está dirigido a cualquier persona interesada en el desarrollo del software AIPA.

1.2. 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.

1.3 Ámbito

- Nombre del software: AIPA.
- El software hará lo siguiente:
 - Tendrá un jugador virtual:
 - Capaz de ir aprendiendo con partidas de ajedrez.
 - Capaz de alcanzar la capacidad de poder vencer a jugadores no expertos.
 - Tendrá un Asistente virtual.
 - Que funcionará como guía del usuario.
- El software no hará lo siguiente:
 - No registrará usuarios.
 - No se conectará con fuentes de datos externas.

- El asistente no hará toma de decisiones, es decir no ejecutará movimientos durante la partida.
 - No enseñara a jugar ajedrez desde 0 (aunque es posible un aprendizaje implícito).
- Los beneficios del software son:
 - Beneficio Investigativo.
 - Entretenimiento.
 - Apoyo a los jugadores de ajedrez.
- Las metas del software son:
 - Especificada en el plan general.

1.4 Referencias

Repositorio del proyecto:

- <https://drive.google.com/drive/folders/1Jq7LTmsebXVyWPEFWUU0gC0e9FrjFzeu?usp=sharing>

2. Descripción del general

2.1 Perspectiva del producto

El software es independiente de otros sistemas, no forma parte de ninguna otra plataforma. Exceptuando lo especificado en suposiciones y dependencias.

2.2 Funciones del producto

El software debe cumplir con las siguientes funciones principales:

- Guiar al jugador en partidas de ajedrez.
- Aprender de partidas de ajedrez.

2.3 Características del usuario

El software está dirigido al siguiente tipo de usuario:

Tipo de usuario	Jugador.
Formación	Con conocimientos de las reglas de ajedrez.
Experiencia en el uso de aplicaciones	Habilidades básicas en el manejo de aplicaciones de escritorio.
Actividad	Jugar al ajedrez

2.4 Restricciones

Las restricciones que tendrá el software son las siguientes:

- El software se desarrollará mediante licencia abierta (open source) y por tanto no se requerirá de licencia para su uso o instalación.
- Debe funcionar sobre una arquitectura mínima de procesador intel i3 de tercera generación con 8 gb de ram.
- Debe funcionar con el sistema operativo windows 10 home de 64 bits.
- Se implementará en lenguaje de programación C#.

Las restricciones que tendrá el desarrollo del software:

De tiempo

- Periodo de desarrollo segundo semestre, Universidad de La Serena.

De habilidad

- Uso de control de versiones git.
- Manejo de lenguaje de programación C#.
- Uso de herramientas inteligentes.

2.5 Suposiciones y dependencias

- Debe ser compatible con el sistema operativo Windows 10.
- La base de datos será local.
- Será monousuario.

3. Requisitos específicos

3.1 Interfaces externas

3.1.1 Interfaces de usuario

- Ventanas.
- Menú.
- Botones.
- Cursor.
- Contenido gráfico.

3.1.2 Interfaces de hardware.

- Teclado.
- Mouse.
- Pantalla.

3.1.3 Interfaces de comunicación

- Base de datos local.
- Archivos.

3.2 Requisitos específicos (por jerarquía funcional)

Especificación de conceptos:

- Transición: Cierre interfaz actual apertura nueva interfaz.
- Apertura: Apertura nueva interfaz.
- Verde: Si la entrada es válida.
- Rojo: Si la entrada no es válida.

Identificación	R.1
Interfaz	Escritorio
Entrada	<ul style="list-style-type: none"> • Ejecutable
Proceso	Se presiona en el ejecutable del software generando apertura del Menú de entrada.
Salida	Bienvenida Mensaje de error

Identificación	R.1.1
Interfaz	Bienvenida
Entrada	<ul style="list-style-type: none"> • Timer
Proceso	Transición a Juego
Salida	Juego Mensaje de error

Identificación	R.1.1.1
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> • Interacción con Botón: Consejos
Proceso	Se presiona el botón Consejos,habilitando actuadores del agente guía.
Salida	Habilita actuadores agente guía Mensaje de error

Identificación	R.1.1.2
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> • Interacción con Botón: Manual de usuario
Proceso	Se realiza apertura del manual de usuario
Salida	Apertura del manual de usuario Mensaje de error

Identificación	R.1.1.3
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> • Resetear
Proceso	Se empieza de 0 la partida de ajedrez
Salida	Resetea el tablero Mensaje de error

Identificación	R.1.1.4
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> • Evento en pieza
Proceso	El jugador clickeo algunas de las piezas del tablero
Salida	Movimientos posibles Mensaje de error

Identificación	R.1.1.1.1
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> Evento generado por el Timer
Proceso	El agente proporciona consejos de movimientos óptimos mediante actualización del label
Salida	Actualización label de consejos. Mensaje de error

Identificación	R.1.1.4.1
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> Evento en movimientos posibles del tablero
Proceso	El jugador clickeo en algunos de los movimientos posibles
Salida	Mueve la pieza a la nueva ubicación. Turno agente. Mensaje de error

Identificación	R.1.1.4.2
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> Evento generado por el timer
Proceso	El agente jugador mueve una pieza del tablero (evidentemente está jugada debe ser racional)
Salida	Mueve una pieza a una nueva ubicación Mensaje de error

Identificación	R.1.1.4.3
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> Estado del tablero.
Proceso	Actualización del estado del tablero según la disposición de las piezas sobre el tablero.
Salida	Estado Mensaje de error

Identificación	R.1.1.4.1.1
Interfaz	Juego
Entrada	<ul style="list-style-type: none"> • Evento en casilla tablero coronación
Proceso	El jugador movió la pieza a una ubicación de coronación generando apertura de la interfaz Coronación
Salida	Apertura Interfaz coronación Mensaje de error

Identificación	R.1.1.4.1.1.1
Interfaz	Coronación
Entrada	<ul style="list-style-type: none"> • Evento en selección de pieza
Proceso	El jugador selecciona una pieza de la interfaz
Salida	Transformación de la pieza peón Mensaje de error

3.3 Requisitos de rendimiento

- El tiempo para aprender una jugada no debe superar más de 1 minuto.

3.4 Requisitos de diseño

Limitaciones del hardware:

- Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:
 - Procesador i3-3120M, CPU @ 2.50GHz o superior.
 - Memoria mínima disponible en el disco de almacenamiento de 256Mb.
 - Memoria Ram: 4GB.

Código:

- Variables: snake case.
- Funciones: Upper snake case.
- Clases: Upper camel case.

Implementación:

- Se utilizará diseño por contrato.
- Se utilizará programación orientada a objetos.

3.5 Atributos del sistema

Especificados en el modelo de calidad AIPA.