

- Code Alchemist

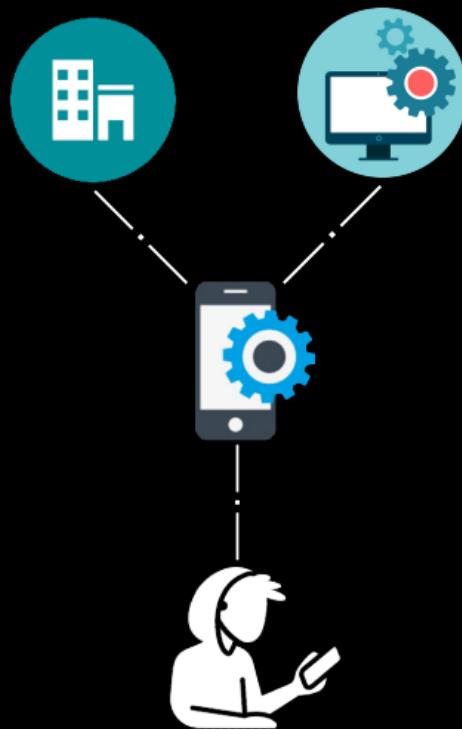
CHECKTER APPLICATION

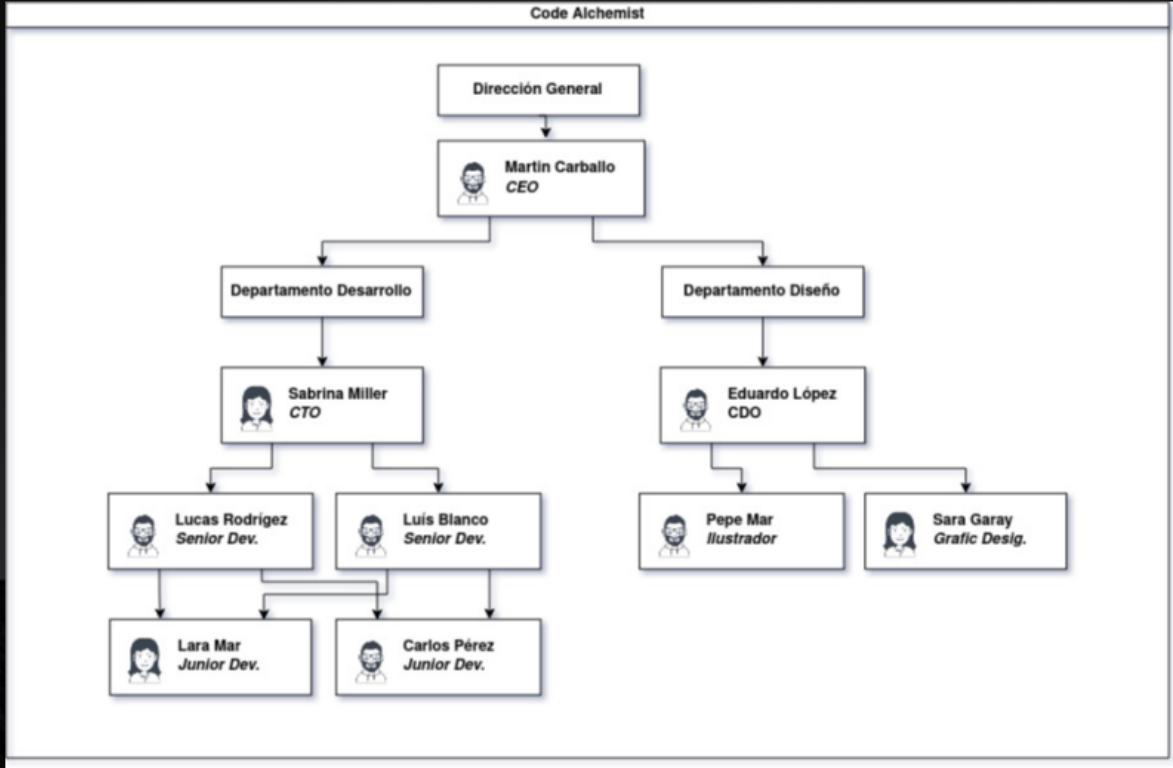


INTRODUCCIÓN

Code Alchemist

- Quienes somos ?
- En que sector trabajamos ?
- Cual es nuestro mercado objetivo ?





Organigrama

TECNOLOGÍAS EMPLEADAS

ÁMBITO	HERRAMIENTAS	CAUSAS
<i>Frontend</i>	  	<i>Multiplataforma</i> <i>Experiencia</i> <i>Desarrollo Ágil</i>
<i>Backend</i>	   Flask 	<i>Simple</i> <i>Desarrollo Ágil</i>
<i>BBDD</i>	  Microsoft® SQL Server	<i>Experiencia</i> <i>Escalabilidad</i>
<i>AI</i>	 Stockfish	<i>Integración Python</i> <i>Extensión de uso</i>
<i>Document.</i>	 LibreOffice 	<i>Experiencia</i> <i>Código Abierto</i>

ARQUITECTURA - MICROSERVICIOS

Ventajas

- Escalable
- Servicios Independientes
- Especialización del equipo
- Flexibilidad

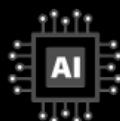
Servicios



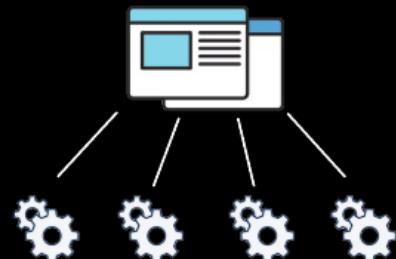
Servicio Aplicación



Servicio Database



Servicio AI



ANÁLISIS DE REQUISITOS

Historias de Usuarios

User 1
★★★★★ 11/6/18

Como jugador ocasional de ajedrez, quiero poder jugar partidas locales contra un amigo o familiares en el mismo dispositivo, para disfrutar del juego cara a cara.

User 2
★★★★★ 11/6/18

Como jugador solitario de ajedrez, quiero poder desafiar a una inteligencia artificial de diferentes niveles de dificultad, para mejorar mis habilidades y disfrutar del juego cuando no tengo compañeros disponibles.

User 3
★★★★★ 11/6/18

Como jugador de ajedrez, quiero poder guardar las partidas que juego, para poder revisarlas más tarde, analizar mis movimientos y aprender de mis errores.

User 4
★★★★★ 11/6/18

Como jugador de ajedrez, quiero poder revisar mis partidas anteriores, para analizar mis movimientos, identificar áreas de mejora y aprender de mis errores.

Requisitos

- Juego local y en linea.
- AI con niveles de dificultad.
- Poder guardar y revisar partidas.

ANÁLISIS DE REQUISITOS

Metodología & Fases temporales del desarrollo

Cascada

- Requisitos bien definidos.
- Publicación completa.
- Desarrollo secuencial.

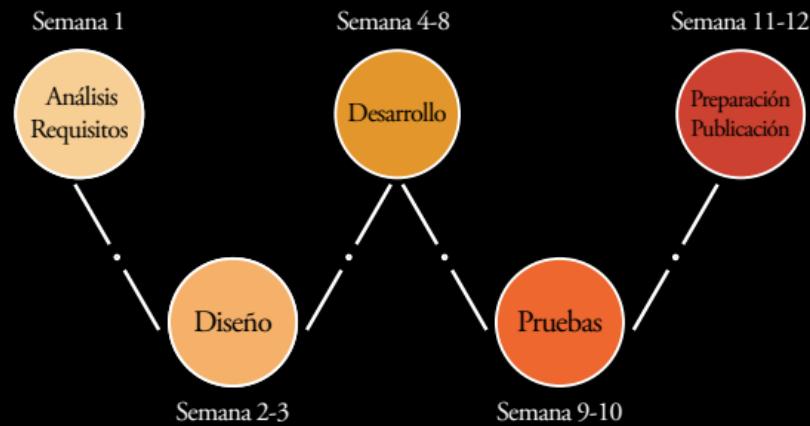
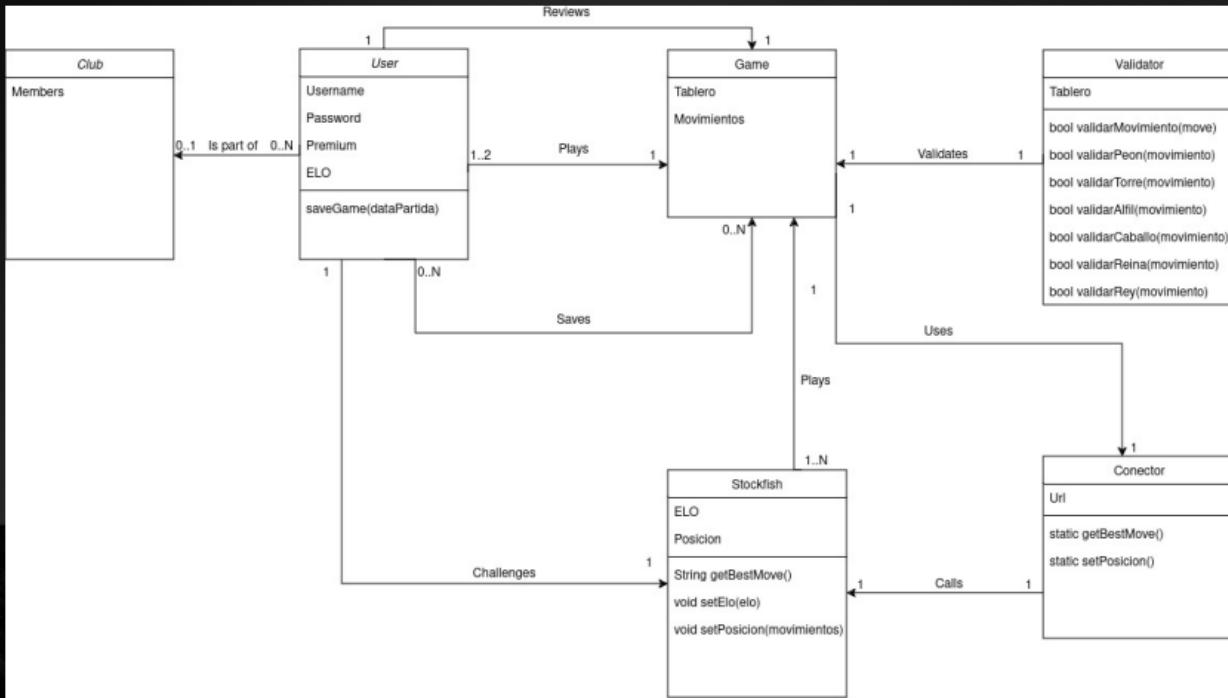
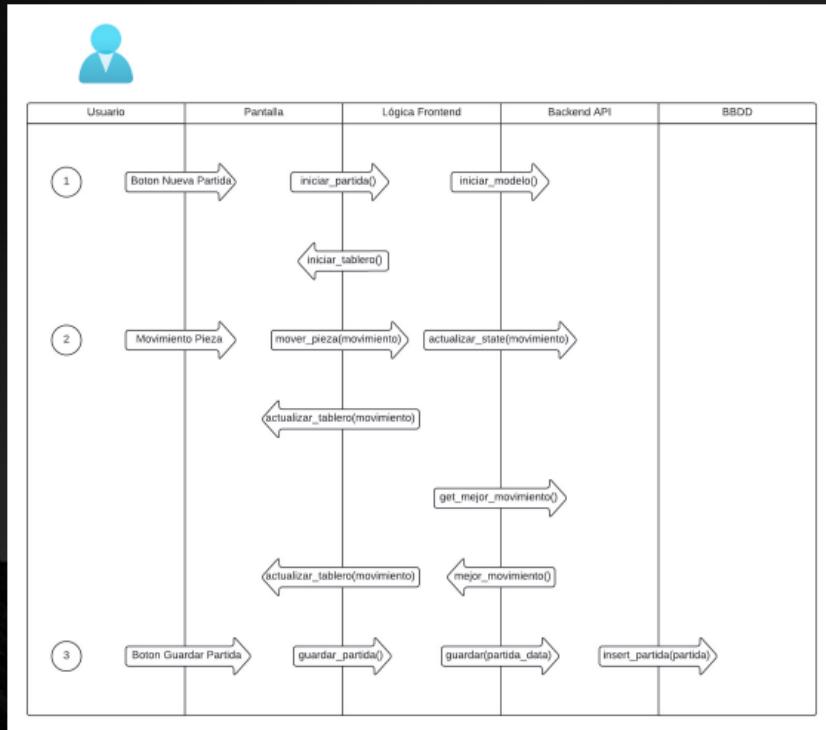


Diagrama de clases UML

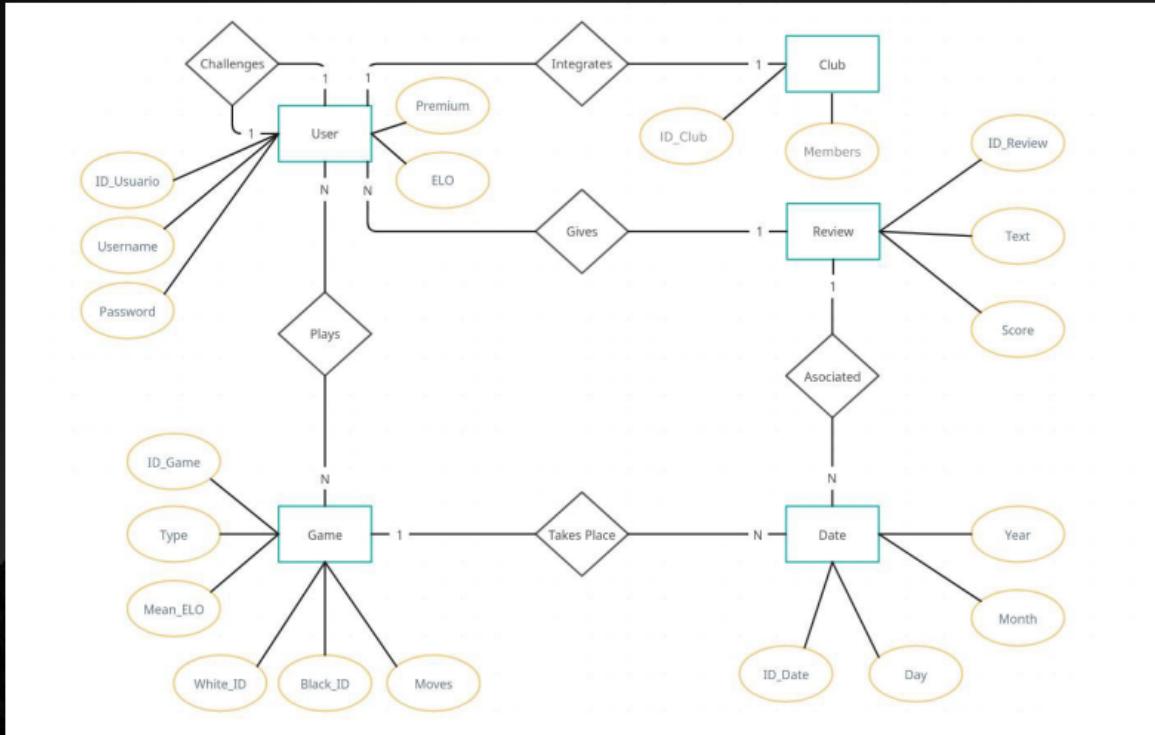


DISEÑO ESTÁTICO



DISEÑO ESTÁTICO

Diagrama de secuencia
de un caso de uso



MODELO DE DATOS

Diagrama ER

MANUALES



Instalación

- Instalación
- Actualización
- Desinstalación



Usuario

- Pantalla Inicio
- Pantalla Menú
- Pantalla Revisión Partida
- Pantalla Partida

VIABILIDAD

Costes & Inversión



Costes

- Salarios - 16182 \$
- Alojamiento Cloud - 450 \$
- Licencias - 150 \$
- Marketing - 5430 \$

Inversión

- Económica - 22212 \$
- Tiempo - Coste de oportunidad de 3 meses de desarrollo.

VIABILIDAD

Previsión esperada

$$(Players\ App\ Store * 0.055 * 10.4 * 0.85) + (Players\ Play\ Store * 0.055 * 1.5 * 0.85) + (Players\ Steam * 0.055 * 1.97 * 0.925) = \text{Ingreso Ventas}$$



Minimos (3000 usuarios)



Máximos (6000 usuarios)



VIABILIDAD

DAFO

PRUEBAS

Pruebas Unitarias

```
def test_guardar_game(self):
    response = self.client.post('/game', data=json.dumps({
        'moves': 'e4 e5 Nf3 Nc6'
    }), content_type='application/json')
    self.assertEqual(response.status_code, 201)
    data = json.loads(response.get_data(as_text=True))
    self.assertIn('message', data)
    self.assertEqual(data['message'], 'Partida guardada.')
    self.assertIn('game_id', data)
```

Pruebas Usabilidad



Pruebas Rendimiento



DEMO



MEJORAS A FUTURO

- 1** - Implementación Login
- 2** - Implementación Review
- 3** - Implementación Configuración
- 4** - Implementación Online
- 5** - Implementación Clubs
- 6** - Nuevos Modelos