

# **GDD Slime Go**



Nombre integrantes: Fernando Castro  
Manuel Cervera  
Ignacio Reyes  
Nicolas Romero  
Gabriel Sepulveda

# Índice

<b>Índice</b>	<b>2</b>
<b>Descripción</b>	<b>3</b>
<b>Jugabilidad</b>	<b>3</b>
<b>Flujo de Juego</b>	<b>4</b>
<b>Experiencia de Usuario:</b>	<b>5</b>
<b>Elementos del juego</b>	<b>7</b>
<b>Arte</b>	<b>11</b>
<b>Cuestionario:</b>	<b>11</b>
<b>Especificaciones Técnicas</b>	<b>12</b>
<b>Referencias</b>	<b>13</b>
<b>Informe Personal</b>	<b>18</b>

## Descripción

- **Nombre producto:** Slime Go
- **Género(s):** Aventura, realidad aumentada
- **Plataforma(s):** Movil , gama media baja o superior.
- **Resumen:**

El objetivo del SlimeGo es crear una experiencia de realidad aumentada interactiva, en la cual el jugador debe capturar diversas criaturas llamadas slimes, recorriendo el mundo en busca de las mismas.

Además de esto el jugador podrá conseguir objetos, encontrar puntos de interés (Paras) y batallar contra diversos slimes, lo que le ayudará a completar su colección

## Jugabilidad

Está subdividida en cuatro categorías, exploración, combate y capturas, la exploración consiste en caminar por el mundo buscando slimes, paradas y objetos, el combate se realiza en tiempo real, pudiendo realizarse diversas acciones, siendo las mismas ataque normal, ataque cargado, esquiva, huida,

y uso de pociones, la captura se puede realizar durante el combate, dependiendo el éxito de la misma de la vida del slime, los ratios se explicaran en mecánicas.

- **Mecánicas principales:**

**Caminar:** Al caminar con el celular en la mano y el juego activo, el jugador es capaz de avanzar dentro del mismo

**Captura:** Durante los combates el jugador puede presionar el botón de captura para intentar capturar al slime, los porcentajes de captura son los siguientes:

- i)  $100\%-50\% = 0\%$  de posibilidad de captura
- ii)  $50\% - 1\% =$  desde un rango del 1 % al 100 % de posibilidades de captura
- iii) Los slimes tienen un 10% de probabilidad de escapar del combate si se falla la captura

**Click and fight:** Al presionar un slime en el mapa se inicia el combate en contra del mismo

- **Mecánicas secundarias:**

**Atacar:** Al tocar la pantalla en combate, se realizan ataques básicos, luego de realizar tres ataques básicos, se puede realizar un ataque cargado dejando presionado el dedo en pantalla

**Esquivar:** Al deslizar el dedo de izquierda a derecha en la pantalla, el slime se mueve rápidamente en la dirección correspondiente, lo que permite esquivar ataques, posterior a este movimiento el slime vuelve a la posición inicial

**Huir:** Al seleccionar el botón de huir en pantalla se puede escapar del combate

**Uso de objetos:** Al seleccionar uno de los botones de poción, el jugador puede curar a sus slimes

## Flujo de Juego

**Inicio:** Al iniciar el juego el jugador debe responder un cuestionario\*, el cual determina su slime inicial.

**Intermedio:** Una vez terminado esto, el jugador puede empezar a recorrer el mundo cazando slimes.

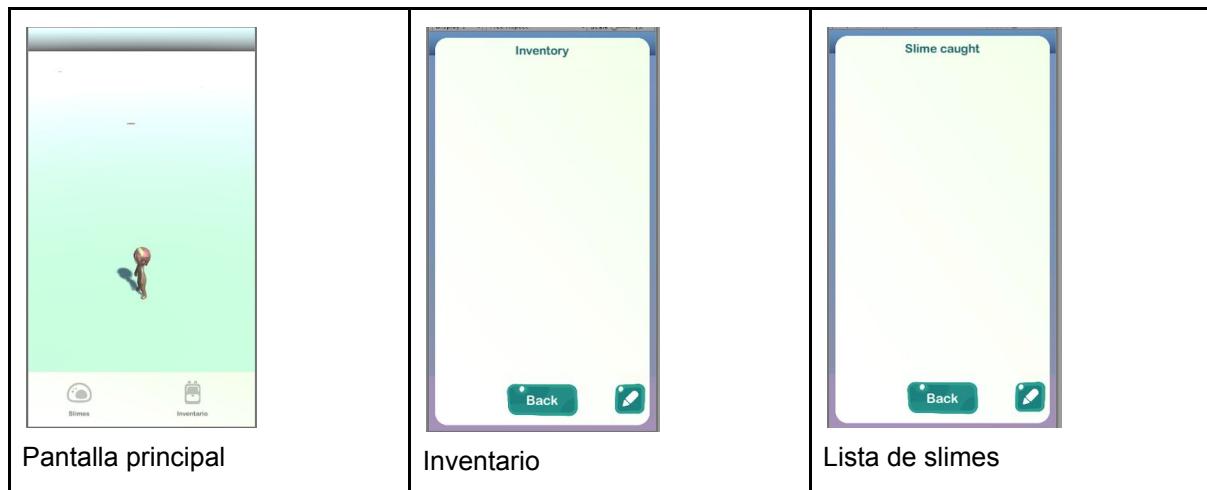
Durante esta fase el jugador luchará contra sus primeros slimes, capturara los mismos, encontrará diversas paradas y aumentará su colección

**Final:** Si bien no existe una fase final como tal, el buscar completar la colección de slimes y perfeccionamiento del equipo se puede considerar como la misma, en esta sección del juego el jugador se dedicara de forma exclusiva a lo anteriormente mencionado, por lo que la duración de la misma es relativa al interés del jugador, ya que el mismo si bien puede completar la colección, puede seguir perfeccionando su equipo buscando mejores slimness el tiempo que desee

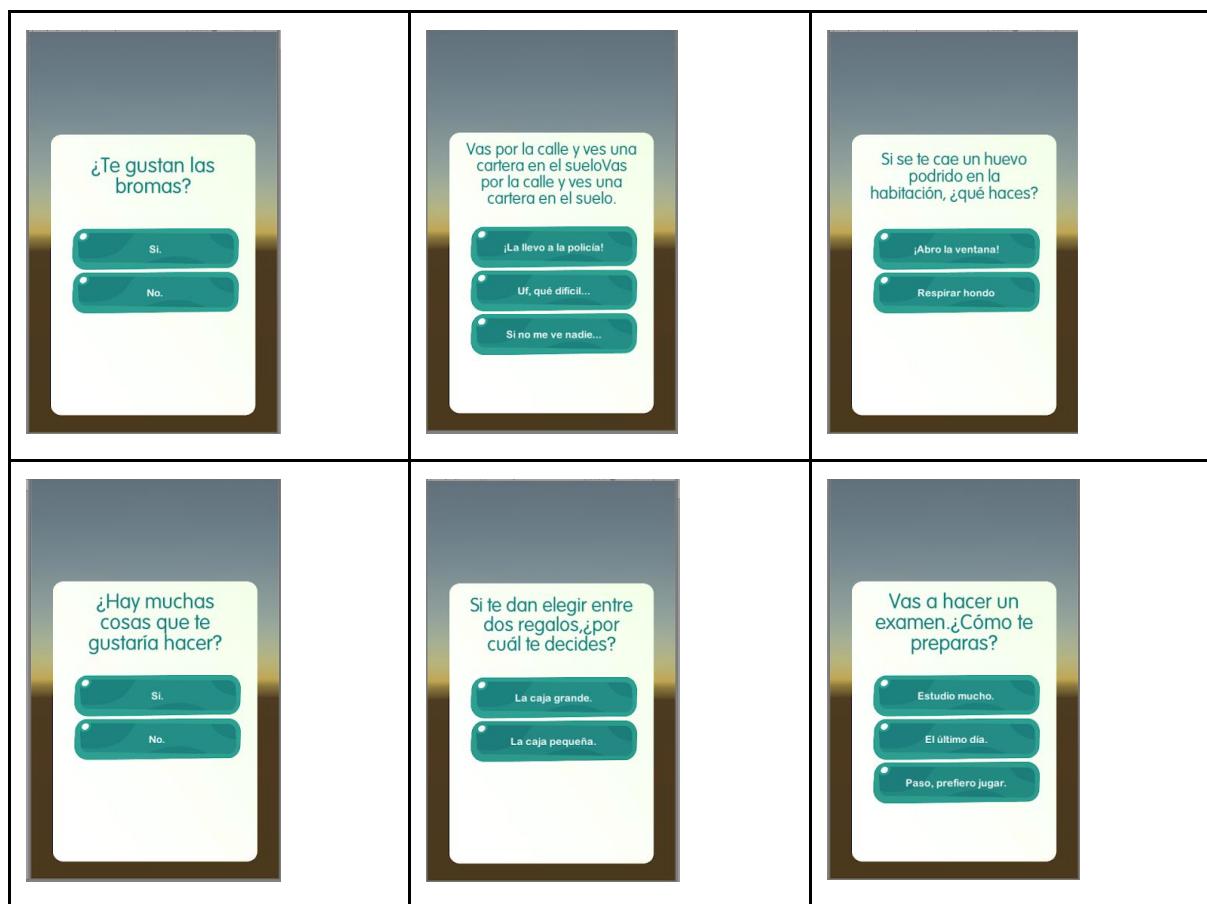
## Experiencia de Usuario:

En esta sección se presentarán todos los apartados visuales relacionados a HUD y menús

HUD:

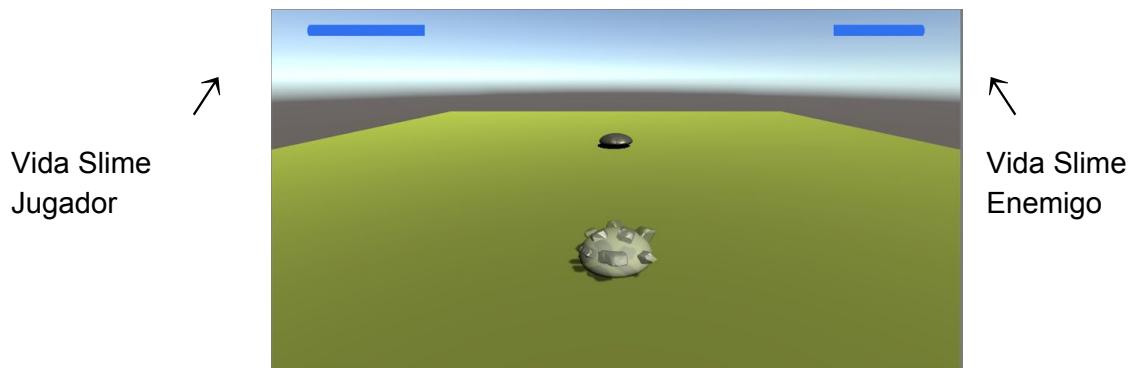


Cuestionario:





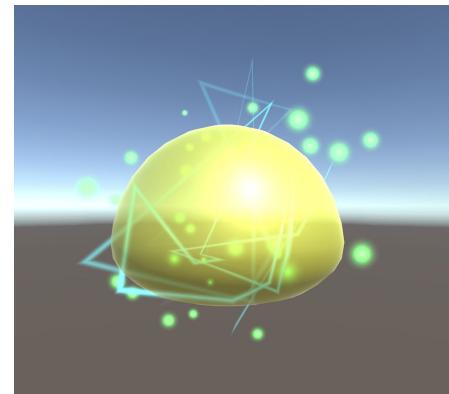
### Escenario de combate:



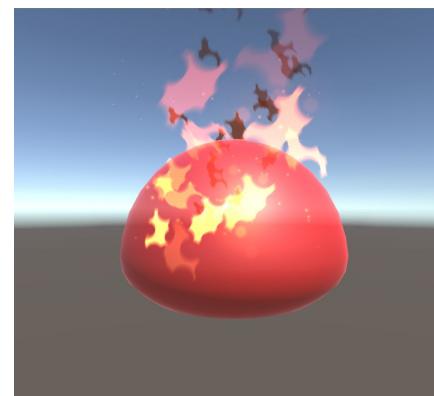
## Elementos del juego

- Lista slimes

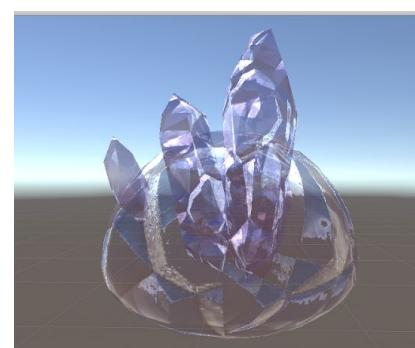
- Eléctrico



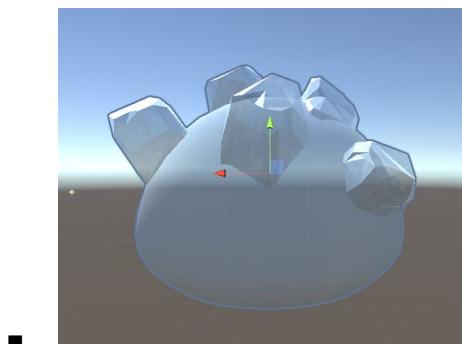
- Fuego



- Gema

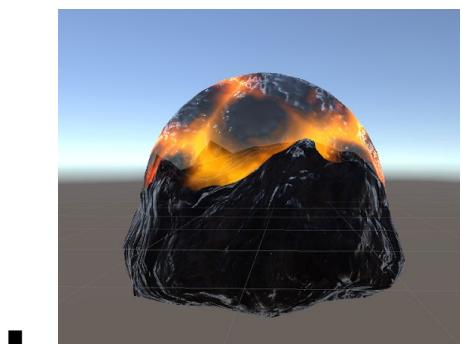


- **Hielo**



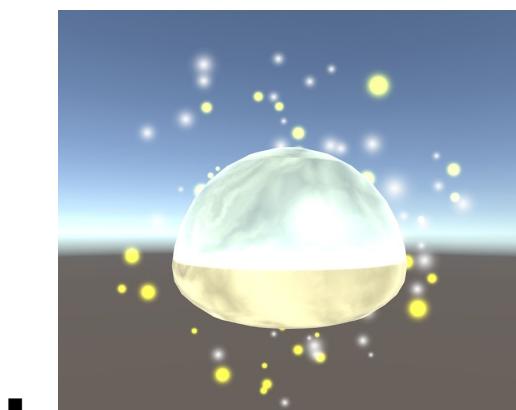
■

- **Lava**



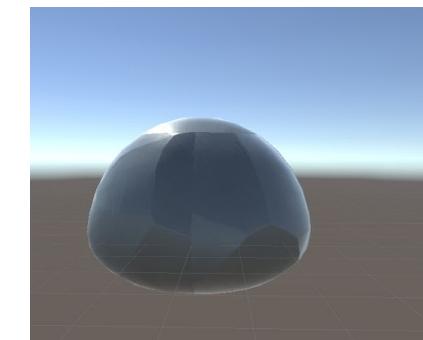
■

- **Luz**

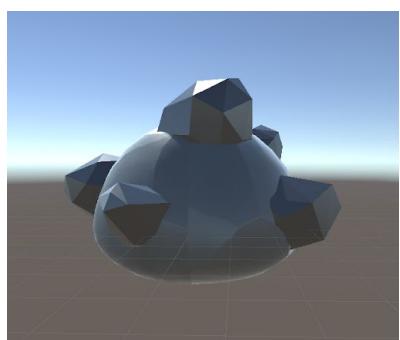


■

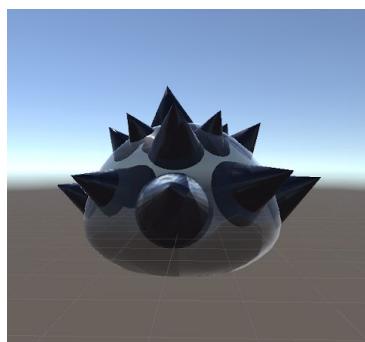
○ **Metal**



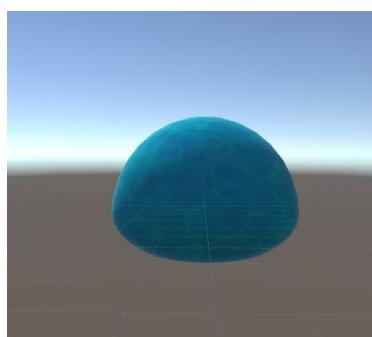
○ **Piedra**



○ **Puas**



○ **Agua**



## Arte

- **Estilo visual:** arte 3D flat. con detalles de partículas y flashes.
- **Especificaciones técnicas:** Modelos 3D flat estilo cartoon.
- **Referencias:** (anexado posteriormente en este documento)

## Cuestionario:

- **Vas a hacer un examen.¿Cómo te preparas?**
  - Estudio mucho.(Piedra)
  - El último día.(Agua)
  - Paso, prefiero jugar(Fuego, Oscuridad)
- **¿Le pones al mal tiempo buena cara?**
  - Sí.(Agua)
  - No.(Electricidad)
- **¿Traicionarias a un amigo?**
  - Sí.(Luz)
  - No.(Oscuridad)
- **Imagina que tienes un cubo.Si tuvieras que llevar agua,¿cómo lo llenarías?**
  - Hasta arriba(Agua)
  - Por la mitad(Electricidad)
  - Un poco(Fuego)
- **Si te dan elegir entre dos regalos,¿por cuál te decides?**
  - La caja grande.(Oscuridad)
  - La caja pequeña.(Luz)
- **Si se te cae un huevo podrido en la habitación, ¿qué haces?**
  - ¡Abro la ventana!(Electricidad)
  - Respirar hondo(Piedra)
- **Un amigo te da algo que se te había olvidado. ¿Cómo se lo agradeces?**
  - Dándole las gracias (Agua)
  - Con una bromilla (Electricidad)
  - Lo normal, sin exagerar (Piedra)
- **Vas por la calle y ves una cartera en el suelo.**
  - ¡La llevó a la policía! (Luz)
  - Uf, qué difícil... (Luz/Oscuridad)
  - Si no me ve nadie... (Oscuridad)
- **¿Has puesto una trampa a alguien alguna vez?**
  - Sí. (Electricidad)
  - No. (Piedra)
- **¿Te gustan las bromas?**

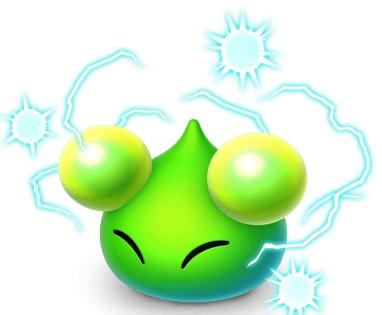
- Sí. (Fuego/Electricidad)
  - No. (Piedra/Agua)
- **¿Hay muchas cosas que te gustaría hacer?**
  - Sí. (Fuego/Piedra)
  - No. (Agua/Electricidad)
- **Un broncas se está metiendo con un amigo. ¿Qué haces?**
  - Me enfrento a él (Fuego)
  - Le amenazó, desde lejos (Agua)
  - Me lanzo a por él (Electricidad)

## Especificaciones Técnicas

- **Game Engine:**
  - Unity:
    - Se escogió debido a que la mayoría de los implicados en el proyecto manejan dicho engine.
    - Facilidad de integración entre el código y el engine.
    - Facilidad a la hora de crear efectos visuales
    - Facilidad de portabilidad a la plataforma objetivo
- **Editor de Código:**
  - Visual Studio
- **Herramientas Creación de Arte Visual**
  - Adobe Illustrator:
    - Versión: CC 2018
    - Uso:
      - Creación de fondos
      - Creación de UI
  - Blender:
    - Modelado de personajes

## Referencias

- Slime eléctrico

	
	<p><b>Características</b></p> <p><b>De color amarillo verdoso con rayos de color azul</b></p>

- Slime de fuego

	
	<p><b>Características</b></p> <p><b>De color rojo con efectos de llamas</b></p>

- Slime Gema

	
	<p><b>Características</b></p> <p><b>Posee gemas que sobresalen de su cuerpo</b></p>

- Slime Hielo

	
	<p><b>Características</b></p> <p><b>De color gris con pinchos de hielo sobresaliente</b></p>

- Slime Lava



**Características**

**De color naranja y negro con líneas que representen la lava**

- Slime Luz



**Características**

**Con un amarillo mucho más tirado al blanco**

- Slime Metal

	
	<p><b>Características:</b></p> <ul style="list-style-type: none"> <li>- Color sólido, textura metálica, reflejos de luz metálicos</li> <li>- partícula de rebote: golpe</li> </ul>

- Slime Piedra

	 <p>colores de este pero forma de ← ese</p>
	<p><b>Características:</b></p> <p>Posee piedras las cuales le sobresalen, cafe</p>

- Slime Púas

	<p><b>Características:</b></p> <p><b>Posee pinchos puntiagudos los cuales le sobresalen</b></p>
---	---

- Slime Agua

 	
	<p><b>Características:</b></p> <ul style="list-style-type: none"> <li>- Azul, celeste transparente</li> <li>- Textura con efecto caustic u oleaje</li> <li>- Gran cantidad de partículas de rebote, azules y pequeñas</li> </ul>

# Informe Personal

Fernando Castro:

- Actividades asignadas:
  - Producción y gestión de actividades
  - Escritura y gestión de documentos (GDD, Carta gant)
  - Animación y modelado del jugador (tanto creación del personaje como Implementación en código)
  - Efectos de sonido y música (tanto búsqueda de sonidos como implementación en código)
- Complicaciones
  - Dificultad para comunicar actividades y cumplir plazos debido al estado actual del país, complicándose esto debido al traslado de uno de los miembros del grupo hacia el extranjero
  - Obsolescencia de los plazos debido cambios de fechas
  - Pérdida de texturas del jugador debido a cambios en el repositorio
  - La necesidad de implementar un sistema que gestiona el total de los sonidos
- Soluciones
  - Se buscó el cumplimiento de los roles asignados en un inicio sin cambiar los mismos, para de esta manera facilitar el entendimiento de los roles individuales, se gestionó el avance de los miembros tanto de forma presencial como no presencial, osea, el uso de herramientas tales como whatsapp o discord.
  - Se adaptaron los plazos cambiando los avances a actividades sencillas, que se pudieran realizar en el tiempo libre, facilitando con esto el completado de las tareas y diluyendo los plazos, los cuales de otra manera no se habían cumplido, se usó el GDD para entregar indicaciones de las características de las actividades
  - Se utilizó el modelo 3D del jugador ya diseñado, agregando la textura de la cara al modelo, por desgracia no fue posible agregar la textura de cuerpo sin realizar de nuevo el proceso de animación, osease la falta de textura en el cuerpo es culpa del presente integrante.
  - Se programaron dos clases, la clase Sound y clase SoundManager, la primera es una clase Serializada que busca guardar la información de los sonidos, siendo la misma nombres, volumen, pitch y Loop, lo que permite que dichas características se pudieran variar de forma sencilla en el SoundManager, en relación a este, su principal función es la gestión de los diversos sonidos y facilitar su uso e implementación

Santiago Cervera:

- Actividades asignadas:
  - Sistema de combate 1v1.
  - Coordinación y arquitectura de código.
- Complicaciones:
  - La necesidad de contar con información específica sobre los combatientes la cual aún no se había desarrollado.
  - Enviar datos de una escena a otra para iniciar el combate con ciertas configuraciones.
  - Coordinacion con mi equipo de trabajo dada la distancia y diferencia horaria.
- Soluciones
  - Procurar tener un sistema lo más genérico posible para poder ser utilizado una vez finalizado el sistema de data de los slimes.
  - Utilizacion de metodos estaticos y coroutines para cargar la escena de manera asíncrona y de esta forma traspasar los datos, considero que existe una mejor opción la cual se me ocurrió al final de la implementación que consiste en la utilización de “scriptable objects” para almacenar los parámetros de la pelea y después cargar la escena de batalla y utilizar esta misma configuración previamente asignada, de esta forma no se tiene que enviar información directamente desde una escena a otra.
  - Actualización del GDD y el uso de herramientas para comunicación grupal tales como discord.

Claudio Rubio:

- Actividades asignadas:
  - Programación y diseño sistema de elección primer Slime en juego basado en test de personalidad(Question System)
  - Programación sistema de inventario para los slimes y para los objetos
  - Programación de shaders visuales de los slimes
  - Integración de interfaces y transiciones entre escenas
- Complicaciones
  - Desconocimiento de la lógica de programación de shaders en unity
  - Modularización de los módulos de inventario y sistema de elección de Slime
- Soluciones
  - Se añaden nuevos shaders sobre los slimes a partir de uno preexistente y se añaden las nuevas funcionalidades necesarias para su correcto funcionamiento
  - Se crean clases auxiliares encargadas de el traspaso de data entre los distintos módulos, uso de namespaces

**Ignacio Reyes:**

- Actividades asignadas:
  - 1.- Diseño de slimes, tipos/elementos y fusiones (en conjunto a más integrantes)
  - 2.- Diseño de spawns de slimes (lugares donde aparece cada slime)
  - 2.- Diseño y modelaje 3D (Slimes / accesorios / proyectiles)
  - 3.- Creación de materiales 3D en Unity
- Complicaciones
  - 1.- Texturas con transparencia tenían errores entre ellas (ejemplo, gemas con transparencia dentro de un slime, también con transparencia), generando una especie de “clipping” al rotarlas y moverlas
  - 2.- Las normales de algunos modelos 3D estaban invertidas al importarlas a Unity desde Blender
  - 3.- La rotación de algunos modelos 3D estaba incorrecta (quedaban rotados por defecto), al importarlas a Unity desde Blender
  - 4.- Generar UV mapping correcto de algunos modelos al usar materiales complejos (ej. material de lava/nieve en slimes)
  - 5.- Adaptación de materiales de blender a Unity (ambos softwares varían en implementación y características de los materiales y texturas)
- Soluciones
  - 1.- Se creó un shader en unity para dejar una renderización fija y que no cambie entre qué objeto debe dibujar antes.
  - 2.- Cambio de parámetros al exportar desde blender
  - 3.- Fixeado localmente desde Unity
  - 4.- Se mejoró el UV mapping de los modelos y offset de los materiales para disminuir los errores
  - 5.- Arreglado mediante investigación de materiales en unity

**Nicolas Romero:**

- Actividades asignadas:
  - 1.- Diseño, creación e implementación de interfaz gráfica.
  - 2.- Busqueda y creacion de modelos 3D de assets secundarios.
  - 3.- Programación del sistema de cuentas y guardado .
  - 4.- Animación de slimes.
  - 5.- creación e implementación de partículas en los slimes.
- Complicaciones
  - 1.- Complicaciones de formato de los modelos en 3D para la implementación de animaciones.
- Soluciones
  - 2.- Creación de animaciones que específicas para los casos particulares en los que se producía la falla.

Gabriel Sepulveda:

- Actividades asignadas:
  - Manejo de Mapa
  - Instanciado de Slimes y objetos en el mapa
  - Gestión de proyecto
  - Pulido de código
  - Tareas de programación misceláneas
- Complicaciones
  - Dado que esta es la primera experiencia que se ha tenido por nuestra parte con Mapbox todo su funcionamiento nos era desconocido. Esto agregó al proyecto la dificultad de aprender a usar la herramienta.
  - El uso de las prestaciones básicas de Mapbox es sencillo, pero al momento de querer realizar ciertas acciones específicas la cantidad de soluciones que existen es limitada y algo difícil de encontrar, esto hizo que el instanciar objetos personalizados en coordenadas específicas del mapa sin que estas tuvieran que estar en código duro se hiciera dificultosa.
  - Dado que el sistema de localización funciona con el GPS del celular no era posible testear de forma cómoda durante el desarrollo de la aplicación, esto se vio agravado dado a que la localización proveída en el editor de Unity no era la que uno asigna manualmente al mapa, si no la que entrega un archivo xml. De esto no existe aviso alguno lo que ocasionó un error del cual no tuvimos pista alguno como resolver o que lo ocasionó.
  - Durante el proyecto se decidió usar un nuevo tipo de arquitectura de programación con un nivel de abstracción mayor al acostumbrado, esto hizo que la creación de código en general fuera un poco más confusa.
  - Dado que varias personas estaban trabajando en diversos sistemas del juego a la vez, y al nivel de intercomunicación necesario entre estos sistemas, ocurrieron casos en los cuales la información no era accesible entre clases.
- Soluciones
  - Se dedicó aproximadamente dos días a lectura de documentación revisión de videos tutoriales y trabajo exclusivo con las herramientas de Mapbox para lograr comprender su funcionamiento y hacer todas las preparaciones necesarias para utilizar el mapa, como la creación del DataSet y la modificación del Data Style.
  - Respecto al tema de la geolocalización en el editor de Unity, se buscó en varios foros de programación hasta que se encontró una respuesta que solucionará el problema.
  - Respecto al tema de la arquitectura de programación, esta se volvió más fácil a medida que se seguía trabajando con ella.
  - Respecto al problema de comunicación entre sistemas, una vez terminados estos fueron revisados y modificados de manera pertinente para que pudieran trabajar como un conjunto.

