



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

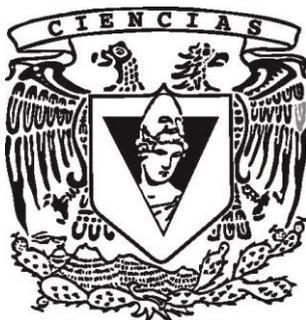
**DESARROLLO DEL SOFTWARE
MULTIMEDIA INTERACTIVO
“EXPLORA LA FUNCIÓN”**

REPORTE DE DIVULGACIÓN

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN**

P R E S E N T A:

IRVING MEDINA LARIOS



**TUTOR:
DR. GUSTAVO DE LA CRUZ MARTÍNEZ**

2017



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno	1. Datos del alumno
Apellido paterno	Medina
Apellido materno	Larios
Nombre(s)	Irving
Teléfono	55-13-44-15-09
Universidad Nacional Autónoma de México	Universidad Nacional Autónoma de México
Facultad de Ciencias	Facultad de Ciencias
Carrera	Ciencias de la Computación
Número de cuenta	304209077
2. Datos del tutor	2. Datos del tutor
Grado	Dr.
Nombre(s)	Gustavo
Apellido paterno	De la Cruz
Apellido materno	Martínez
3. Datos del sinodal 1	3. Datos del sinodal 1
Grado	Dra.
Nombre(s)	Hanna
Apellido paterno	Jadwiga
Apellido materno	Oktaba
4. Datos del sinodal 2	4. Datos del sinodal 2
Grado	L.D.G
Nombre(s)	Ana Libia
Apellido paterno	Eslava
Apellido materno	Cervantes
5. Datos del sinodal 3	5. Datos del sinodal 3
Grado	Ing.
Nombre(s)	Ricardo
Apellido paterno	Castañeda
Apellido materno	Martínez
6. Datos del sinodal 4	6. Datos del sinodal 4
Grado	Dra.
Nombre(s)	Selene Marisol
Apellido paterno	Martínez
Apellido materno	Ramírez

7. Datos del trabajo escrito.

Título

Subtítulo

Número de páginas

Año

7. Datos del trabajo escrito.

Desarrollo del software multimedia interactivo “Explora la función”

75 p

2017

Contenido

Introducción	3
Motivación	3
Organización	4
1 Ingeniería de software	5
1.1 Procesos de software	5
1.2 Procesos Iterativos.....	8
1.2.1 Desarrollo incremental	9
1.2.2 Desarrollo evolutivo	9
1.3 Modelo de desarrollo concurrente.....	12
1.4 Resumen.....	13
2 Proceso de desarrollo de software multimedia interactivo	14
2.1 Roles.....	17
2.2 Planteamiento y análisis	18
2.3 Diseño.....	20
2.4 Realización	22
2.5 Pruebas y evaluación.....	24
2.6 Cierre.....	26
2.7 Retroalimentación y mantenimiento	28
2.8 Resumen.....	30
3 Aplicación del proceso de desarrollo de software multimedia interactivo	31
3.1 Planteamiento y análisis	31
3.1.1 Tema del interactivo	31
3.1.2 Objetivos.....	32
3.1.3 Perfil de usuario.....	32
3.1.4 Contexto de uso	32
3.1.5 Descripción del interactivo	32
3.2 Diseño.....	35
3.2.1 Diseño de propuestas gráficas	36
3.2.2 Estilo gráfico.....	37

3.2.3 Descripción del funcionamiento	39
3.2.4 Productos generados	45
3.3 Realización	48
3.3.1 Arquitectura de Explora la función	48
3.3.2 Organización de la línea de tiempo	50
3.3.3 Animaciones	54
3.3.4 Código	55
3.3.5 Desempeño	60
3.3.6 Contenidos	61
3.4 Pruebas	62
3.4.1 Evaluación con usuarios	62
3.4.2. Evaluación y análisis de los expertos	63
3.5 Cierre	65
3.6 Retroalimentación y Mantenimiento	67
3.7 Resumen	68
4 Conclusiones	69
4.1 Resumen	69
4.2 Observaciones finales	70
4.3 Trabajo a futuro	71
Anexo A	72
Bibliografía	75

Introducción

Hoy en día es muy común escuchar la palabra software, pero realmente qué significa y qué alcance tiene este término. Entendamos software como algo intangible, algo que sabemos que existe pero que es imposible verlo físicamente.

Podemos definir al software como un conjunto de instrucciones para realizar una tarea específica que deberá ejecutar una computadora. Con la ayuda de los lenguajes de programación, que paradójicamente también forman parte del software, podemos generar dichas instrucciones y transformarlas en un programa de computadora.

En contraparte del software, el hardware, como algo que es meramente físico, tiende a desgastarse a través del tiempo, pero dado los grandes avances en este ramo, es posible reemplazar una pieza de hardware sin problema alguno. En el caso del software el mantenimiento es mucho más complejo, el software no se desgasta, pero sí cambian las necesidades de la tarea, por eso la importancia de un buen diseño que permita realizar cambios a los sistemas de software.

Motivación

La creación de software siempre ha sido el área de mayor interés para mí, en particular el ámbito de la programación, darle instrucciones a una máquina para que lleve a cabo una actividad es lo que más me gusta de mi carrera. Con base en esto no dude en aceptar la oportunidad para conocer el proceso de desarrollo de software multimedia interactivo, primero como estudiante en un seminario en la Facultad de Ciencias y después, de una manera más formal, trabajando con el grupo Espacios y Sistemas Interactivos para la Educación (ESIE) del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET). Colaborando con el grupo ESIE, tuve la oportunidad de crear las animaciones y ser parte del equipo de programadores que implementaron el interactivo llamado: *Explora la función*, el cual es parte de las exhibiciones del museo Papalote Verde de Monterrey.

Cualquier proyecto de desarrollo profesional de software requiere de un diseño que permita darle mantenimiento y realizar los ajustes que requiera el proyecto, en particular para *Explora la función* es fundamental una documentación que permita que los nuevos programadores y animadores sean capaces de entender la solución actual. En este trabajo se presenta todo lo relacionado al desarrollo del proyecto y el proceso que se sigue, así que puede ser usado

como referencia para que los nuevos integrantes del equipo comprendan el proceso y les sea más fácil la integración al equipo de trabajo. Adicionalmente, en este trabajo se hace un análisis del proceso de desarrollo que sigue el grupo ESIE, con el fin de mejorarlo para el desarrollo de próximos proyectos.

Organización

En el capítulo 1 de este trabajo, se dará un panorama general de lo que es la ingeniería de software, se describirán los principales modelos de los procesos de software usados en la actualidad, para posteriormente en el capítulo 2, explicar las características del proceso de software utilizando por el grupo ESIE.

La comprensión de este proceso por parte de todo el equipo de desarrollo es de gran relevancia para la construcción de software multimedia interactivo que cumpla los objetivos planteados por el cliente. Este es el caso del software *Explora la función*, por lo cual en el capítulo 3, se hablará a fondo de cómo se realizó el desarrollo de este interactivo.

Finalmente, en el capítulo 4, se presentarán las conclusiones donde se hablará de las ventajas y desventajas del proceso, así como las observaciones y lecciones aprendidas, además de mencionar el trabajo a futuro que se tiene planeado para el interactivo.

1 Ingeniería de software

Para garantizar que el software sea de calidad existe la ingeniería de software, la cual define un marco de trabajo basado en procesos, métodos y herramientas que permitan realizar en tiempo y forma el desarrollo, la operación y el mantenimiento del software de computadora (Pressman, 2005).

El punto principal de la ingeniería de software es el proceso, definamos al proceso como un conjunto de actividades relacionadas entre sí, que se llevan a cabo a través de ciertos roles específicos asignados a elementos automatizados cuyas acciones conducen a la creación de un producto de software (Oktaba, et al., 2005). El proceso permite al equipo de trabajo saber qué es lo que se tiene que hacer, cuándo, cómo y quién es el responsable de hacerlo (Ibargüengoitia y Oktaba, 2010).

Un rol define las responsabilidades a cumplir dentro del proceso y describe una serie de tareas asignadas a una o más personas. Una persona puede tener uno o más roles.

Los métodos constituyen los aspectos técnicos necesarios para facilitar el desarrollo de software, representan un amplio conjunto de soluciones para las tareas que se necesitan hacer en el análisis de requisitos, el diseño, la construcción y la validación de calidad del software.

Las herramientas dan el soporte para el proceso y los métodos, representan los insumos necesarios para la creación del software.

1.1 Procesos de software

No existe un proceso ideal que pueda ser aplicable a todos los tipos de software que existen hoy en día, sin embargo es posible definir un proceso general que plantee una serie de actividades fundamentales para el desarrollo de software como las siguientes:

- Especificación de requerimientos: Establecer comunicación entre cliente y equipo de trabajo con el fin de establecer los requisitos, funcionalidad y restricciones del software.
- Diseño e Implementación del software: Diseñar la estructura que tendrá el software con ayuda de métodos y herramientas para después implementarlo en algún lenguaje de programación.
- Validación del software: Se corrobora que el software no contenga errores de programación y diseño con el fin de que se esté cumpliendo con todas las peticiones del cliente.
- Evolución del software: La entrega del software de forma completa o parcial es evaluado por el cliente y a partir de la retroalimentación se modifica o expande el producto con base en los cambios solicitados.

En la práctica resultan ser actividades complejas que incluyen un conjunto de tareas (Sommerville 2011).

Cada tarea es asignada a una o más personas que llevarán un rol específico durante el proceso y cuya función será ejecutar y completar dichas tareas para alcanzar las metas del desarrollo.

Las organizaciones o equipos de trabajo encargados del desarrollo de software deben tener sus actividades definidas para el proceso de software que adopte y adecuarlo a la naturaleza específica de cada proyecto, esto implica adecuar el proceso tomando como referencia un modelo, el modelo de un proceso no es más que una representación abstracta de un proceso. Dichos modelos no son descripciones finales de los procesos de software, son más bien una secuencia de actividades que permiten heredar y extender sus características para crear procesos más específicos de la ingeniería de software.

Los modelos más usados en la actualidad toman como base las actividades del proceso descrito anteriormente y se describen de una forma muy general de la siguiente manera:

- **Modelo en cascada (Sommerville, 2011):** Toma las actividades genéricas del proceso descrito anteriormente y las representan con fases separadas.
 - Análisis y definición de requerimientos: El objetivo, los servicios y las limitaciones del sistema se definen a detalle mediante entrevistas con el cliente con el fin de tener una completa especificación del sistema.
 - Diseño del software: Se diseña el sistema, se establece los requerimientos para el hardware y el software y se define como será la funcionalidad del software con base en la especificación del sistema.
 - Implementación: El diseño del software se representa en un conjunto de programas mediante uno o más lenguajes de programación.
 - Pruebas de calidad: El conjunto de programas generados en la implementación se integran como un sistema completo y se evalúa para garantizar que no haya errores y que se esté cumpliendo con los requerimientos para después liberar el sistema de software al cliente.
 - Operación y mantenimiento: El sistema se instala y se pone en práctica. El mantenimiento se lleva a cabo si se detectan errores que no fueron resueltos en las etapas anteriores o si se descubren nuevos requerimientos necesarios para mejorar el sistema.

Generalmente no se puede empezar una actividad hasta que la actividad previa haya sido concluida, aunque en la práctica las actividades se traslapan, ya que en el diseño pueden identificarse problemas con los requerimientos o en la implementación se detectan problemas con el diseño y así sucesivamente. Es por esto que no se representa de una manera del todo lineal ya que se necesita realizar iteraciones al modelo dado la fuerte retroalimentación que hay entre sus actividades.

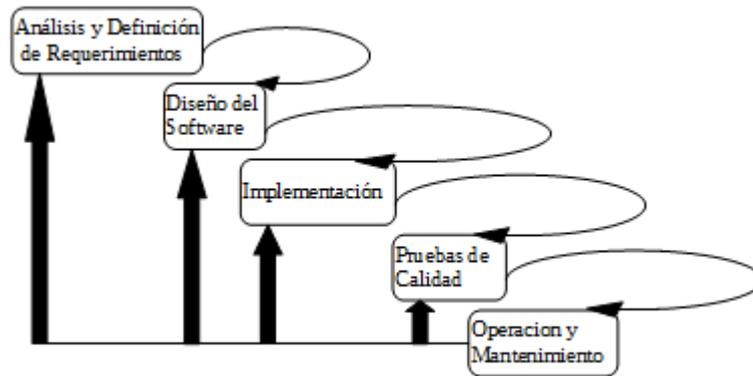


Figura 1.1 Modelo en Cascada.

Es muy raro que los proyectos de software reales puedan llevar una secuencia lineal como lo propone el modelo en cascada además de que es necesario tener todos los requerimientos del software desde un inicio y eso es algo difícil de lograr por el cambio constante al que se suele someter el software.

- **Modelo incremental o iterativo:** Se desarrolla una versión de software funcional con base en especificaciones generales que aporte el cliente, en dicha versión se cumple solo con las necesidades básicas de los requerimientos del producto. Esta versión se va incrementando y perfeccionando en cada nueva versión del software. Los modelos iterativos suelen mezclarse con el modelo en cascada.

Estos modelos son los más usados en la actualidad y serán descritos más a fondo en la siguiente sección dada su importancia para fines del presente trabajo.

- **Modelo de software orientado a la reutilización (Sommerville, 2011):** Cuando el equipo de trabajo ya ha hecho códigos o diseños similares a los requeridos por el cliente, estos componentes pueden ser reutilizables, el enfoque de este modelo se centra más en la integración de estos componentes que en desarrollarlos desde cero.

En este modelo las actividades de definición de requerimientos y pruebas son equivalentes a las descritas en los modelos anteriores, la diferencia radica en sus actividades intermedias y estas son las siguientes:

- **Análisis de componentes:** Con base en los requerimientos del sistema, el equipo de trabajo se encarga de buscar componentes que puedan ayudar a cumplir con la especificación del software a realizar, difícilmente los componentes son iguales a los del sistema requerido así que los que pudieran ser útiles representan solo una parte de la funcionalidad requerida.
- **Modificación de requerimientos:** Se analizan los requerimientos con base en los componentes identificados como útiles, después se modifican de acuerdo a los componentes disponibles, en los casos donde los cambios no sean factibles se puede regresar a la actividad de análisis de componentes.

- Diseño de sistema con reutilización: El equipo de trabajo cuenta con los componentes que pueden ser reutilizables con base en esto se diseña el concepto del sistema o se reutiliza un concepto ya existente. Si no fue posible encontrar todos los componentes para completar las especificaciones del software estas se tienen que diseñar desde cero.
- Desarrollo e Integración: Se desarrolla el software que no fue posible reutilizar y que es necesario para cumplir la especificación del sistema para después integrarse con los componentes disponibles.

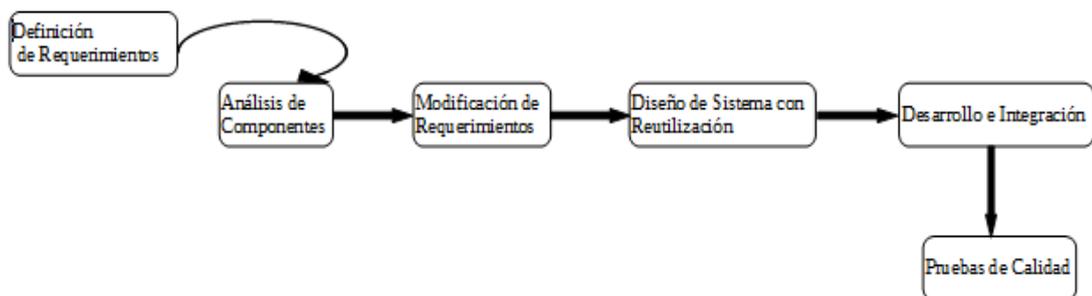


Figura 1.2 Modelo de Software Orientado a la Reutilización.

La gran ventaja de este modelo es que el código a desarrollar se reduce considerablemente debido a la reutilización de componentes lo que implica una disminución de costo y tiempo. Sin embargo la modificación de requerimientos hace que se pueda generar un software que no cumpla con las necesidades reales del cliente.

Estos modelos de procesos son genéricos y no son excluyentes uno del otro, ya que en la práctica generalmente se combinan.

1.2 Procesos Iterativos

El modelo en cascada representa el paradigma clásico y a su vez el más sencillo dada su secuencia lineal, sin embargo cuenta con ciertas deficiencias, ya que al ser secuencial es necesario esperar el término de una actividad para comenzar una nueva, esto provoca estados de bloqueo que se traduce en pérdida de tiempo para el equipo de trabajo. La necesidad de modificaciones en el software son un punto más en contra para el modelo en cascada, ya que al surgir cambios espontáneos que no se contemplaron en el diseño o en los requerimientos del sistema, suelen posponerse para después o improvisar una solución mediante programación lo que puede provocar errores en el software o que el sistema no haga lo que el cliente quiere.

Con base en dichas deficiencias del modelo en cascada surgieron los procesos iterativos, que son modelos capaces de responder al cambio que suele ocurrir durante el desarrollo del software, permitiendo repetir regularmente las actividades conforme el sistema evoluciona, ya sea por petición del cliente o por mejoras al software. El software no se desarrolla en su totalidad sino en una serie de incrementos, dando en cada incremento nuevas funcionalidades al sistema.

Algunos de los modelos que siguen el paradigma iterativo se mencionan a continuación:

1.2.1 Desarrollo incremental

Se define un conjunto de requisitos básicos y de prioridad más alta para el software y basándose en esto se trabaja una versión inicial, a partir de ahí se va refinando en nuevas entregas. Es una combinación del modelo en cascada aplicado en una forma iterativa, con base en una primera secuencia lineal del modelo en cascada se incrementa de manera escalonada una nueva secuencia con el fin de que en cada secuencia agregada haya un incremento de software, es decir, partimos de un producto básico que el cliente obtiene en una primera entrega y, con base en sus comentarios, se desarrolla en una nueva iteración otra secuencia lineal para incrementar su funcionalidad, el proceso se repite hasta que el software haya sido desarrollado por completo (Pressman, 2005).

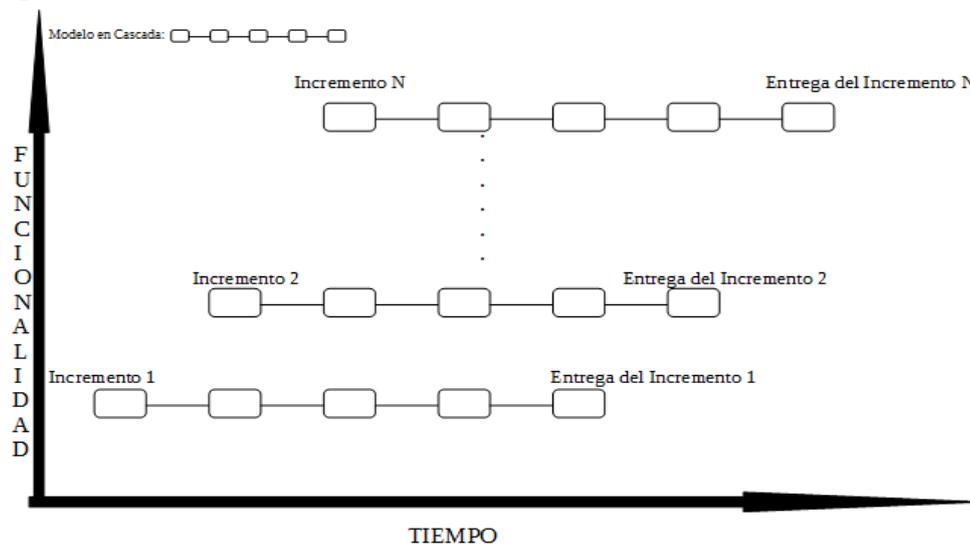


Figura 1.3 Modelo Incremental.

Una de las ventajas del modelo incremental es que el cliente siempre tiene una versión funcional de su software y pueden experimentar con él, lo cual les permite clarificar requerimientos para los incrementos posteriores. Hay que tomar en cuenta que, para evitar posibles retrasos, los incrementos no deberían ser muy grandes y cada incremento debería de permitir crear una nueva funcionalidad al sistema.

1.2.2 Desarrollo evolutivo

En muchas ocasiones el cliente mismo no tiene claro ni definido que es lo que realmente quiere que haga su software, tiene una idea vaga y abstracta de lo que desea. Esto implica una serie de cambios a futuro para el producto en cuestión y la necesidad de usar un proceso que permita al software evolucionar a través del tiempo.

Tomando como base los requisitos mínimos y la idea general del software, se empieza a desarrollar una versión inicial que se expone a los comentarios del cliente y con

base en ello se va mejorando en versiones posteriores, hasta que llegue a ser un sistema acorde a las peticiones del cliente.

Las actividades de especificación, desarrollo y pruebas se relacionan aún más, para que la retroalimentación aclare la funcionalidad y los alcances del producto.

Algunos tipos de desarrollo evolutivo son:

1.2.2.1 Construcción de prototipos

Este paradigma tiene como finalidad comprender y definir los requerimientos del sistema con el cliente, mediante ideas globales y requisitos básicos se plantea de manera rápida una propuesta que lleva a la construcción de un prototipo que permita la interacción con el usuario, para que con base en ello se pueda comprender mejor los requerimientos del software, probar opciones de diseño, mostrar conceptos y a su vez, evolucionar de un prototipo burdo a un producto que llegue a ser lo que el cliente desea.

El modelo de prototipos (Pressman, 2005) comienza con la actividad de comunicación con el cliente, en donde se definen las ideas globales del software, a partir de esto, se realiza una iteración de construcción de prototipos mediante el modelado de la tarea y diseño rápido se representan características visibles para el cliente por ejemplo el diseño de la interfaz de usuario.

El diseño llevara a la construcción de un prototipo para la evaluación del cliente y obtener una retroalimentación que defina la forma en que será el desarrollo final del sistema.

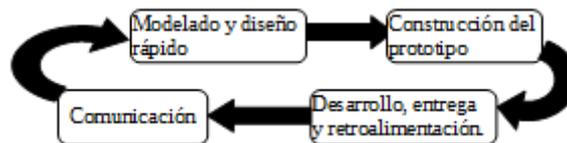


Figura 1.4 Modelo de Prototipos.

Conforme se desarrolla el prototipo se visualizan nuevas ideas, debilidades y fortalezas del software que permiten tanto al usuario y al cliente definir requerimientos más exactos para el desarrollo del sistema. Una de las desventajas de este tipo de desarrollo es que, al ser necesario crear un prototipo de manera rápida, se suele dejar de lado la calidad del código, ya que los programadores en un afán de crear algo funcional pueden implementar código ineficiente. Es por eso que el modelo basado en prototipos tiene que ser exclusivo para la definición de requisitos para después desarrollar el software real con un enfoque hacia la calidad.

1.2.2.2 Modelo en espiral

El modelo de Boehm (Sommerville, 2011), es un proceso de software dirigido por el riesgo, el proceso se representa como un espiral, cada ciclo del espiral representa una actividad del proceso, es decir, el ciclo más interno puede representar la factibilidad

de realizar el sistema, el siguiente ciclo los requerimientos, el siguiente ciclo el diseño, el siguiente el desarrollo y así sucesivamente.

Al ser un modelo evolutivo, asume que en cada cambio hay un riesgo para el proyecto e incluye actividades de gestión para reducir dichos riesgos.

Las principales actividades del modelo de Boehm son:

- Establecimiento de objetivos: Se detectan los objetivos específicos, las restricciones de la actividad en curso y los riesgos del proyecto, para luego realizar un plan de administración detallado y poder plantear estrategias con base en los riesgos.
- Valoración y reducción de riesgo: A cada riesgo detectado se le realiza un análisis para definir acciones que puedan evitarlo.
- Desarrollo y validación: Con base en el análisis de riesgos se escoge el modelo de software más acorde para el sistema, se realiza el desarrollo y después se ejecutan pruebas de validación.
- Planeación: Se realiza una revisión del proyecto para decidir si se puede avanzar a un nuevo ciclo de la espiral y trazar los nuevos objetivos.

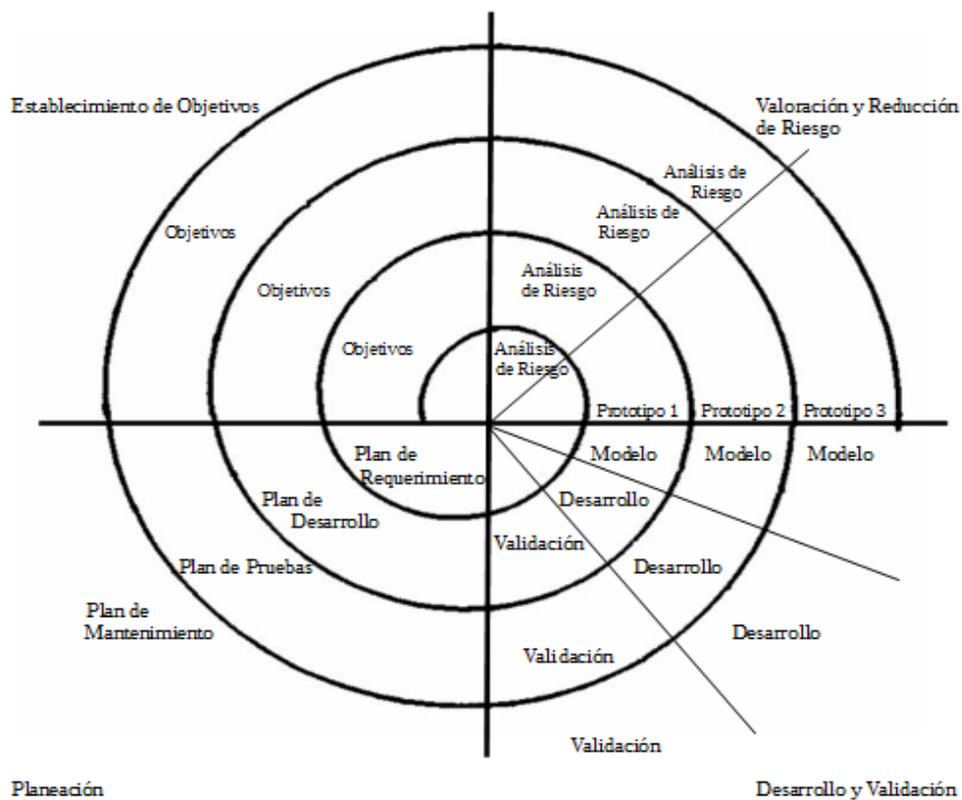


Figura 1.5 Modelo en Espiral.

El software se desarrolla como una serie de entregas evolutivas, las primeras entregas puede ser desde algo sencillo como un documento o un prototipo, hasta llegar a las últimas iteraciones que contiene versiones cada vez más cercanas al producto final.

1.3 Modelo de desarrollo concurrente

Se representa por una serie de actividades y tareas asociadas a un estado. Todas las actividades existen de manera concurrente pero tienen distintos estados. En base a una serie de eventos se puede realizar una transición en el estado de las actividades, por ejemplo en un inicio como primera iteración podríamos tener en la actividad de comunicación con el cliente un estado de “en proceso” y en la actividad de planeación y desarrollo un estado de “ninguno”, ya que al ser la primera iteración del proceso no hay aún requerimientos para comenzar a trabajar. Una vez definido los requerimientos podemos cambiar la actividad de planeación a un estado de “en proceso” y la actividad de comunicación a un estado “en espera de cambios”. Es aplicable a todos los desarrollos de software ya que presenta una visión actualizada del estado de todas las actividades y por ende saber el estado exacto del proyecto (Pressman, 2005).

La imagen 1.6 toma como ejemplo una tarea relacionada con la actividad de implementación para el modelo de desarrollo concurrente.

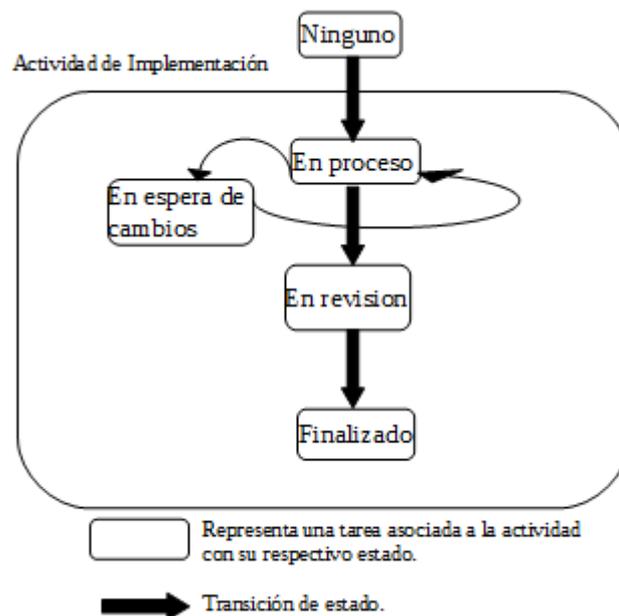


Figura 1.6 Modelo de Desarrollo Concurrente.

En cada actividad puede haber distinto número de estados para sus respectivas tareas y, dependiendo del estado que tengan, pueden generar eventos que cambien el estado ya sea de las tareas o de las actividades (Pressman, 2005).

Estos modelos iterativos representan una solución al cambio constante del software y en general cada uno de ellos cuenta con características de algún otro. El desarrollo iterativo es de cierta manera el enfoque más usado para el desarrollo de sistemas y es precisamente este tipo de modelo el usado en el proceso de desarrollo de software multimedia interactivo.

1.4 Resumen

En este capítulo se revisaron diversos conceptos relacionados con la ingeniería de software que nos ayudan a tener un panorama general de este campo. En particular, serán relevantes para nosotros las siguientes ideas: la ingeniería de software como un marco de trabajo basado en métodos, herramientas y procesos; los métodos como aspectos técnicos necesarios para ayudar al desarrollo de software; las herramientas como el soporte para el proceso y los métodos.

De esta manera, entenderemos que un proceso es un conjunto de actividades relacionadas entre sí; dichas actividades se realizan a través de roles asignados a personas que al ejecutarlas en conjunto desarrollan un producto de software. De acuerdo con la discusión presentada en este capítulo, no existe un proceso que pueda ser perfecto para todo tipo de software pero a partir de los modelos generales se pueden definir procesos específicos que se ajustan a problemáticas y necesidades concretas, uno de esos procesos es el que se sigue en el desarrollo de este proyecto y se presentará en el siguiente capítulo.

2 Proceso de desarrollo de software multimedia interactivo

“El término multimedia interactivo hace referencia al uso combinado de diferentes medios de comunicación: texto, imagen, sonido, animación y video”. (Belloch, pág. 1) Los programas de computadora que utilizan de forma combinada y coherente para sus objetivos dichos medios y permitan la interacción con el usuario se llamarán software multimedia interactivo. Este proceso está enfocado en el desarrollo de dicho software para ser usado en distintos campos tales como la divulgación, educación, juegos, arte o publicidad.

El proceso utilizado para el desarrollo de software multimedia interactivo seguido por el grupo ESIE del CCADET de la Universidad Nacional Autónoma de México (UNAM), es una combinación del modelo incremental y evolutivo mencionados en el capítulo anterior, ya que durante el desarrollo del software se hacen entregas incrementales y además la creación de prototipos para la realización del software.

Se compone por seis actividades o fases siguientes:

- Planteamiento y análisis
- Diseño
- Realización
- Pruebas y evaluación
- Cierre
- Retroalimentación y Mantenimiento

Para desarrollar las distintas fases, el equipo de trabajo divide las tareas en tres áreas para llevarlas a cabo concurrentemente. Las áreas son las siguientes:

- Administrativa: El objetivo general del área es mantener el software multimedia interactivo bajo una planeación adecuada, asume el control de riesgos y permite monitorear el software para saber el estatus de este.
- Desarrollo: Se enfoca en diseñar e implementar un software multimedia interactivo de calidad.
- Control de calidad: Se encarga de garantizar que el software cumpla con todos los requerimientos y satisfaga las peticiones del cliente mediante distintos niveles de pruebas de calidad.

Cada área organiza un conjunto de roles para crear equipos de trabajo, dichos equipos tendrán asignadas tareas en común. Un rol puede estar en uno o más equipos de trabajo.

	Área Administrativa	Área de Desarrollo	Área de Control de Calidad
Planteamiento y análisis	Planteamiento y análisis	Concepción	Aseguramiento de calidad del planteamiento
Diseño	Planeación	Diseño	Aseguramiento de calidad del diseño
Realización	Seguimiento de la producción	Realización	Aseguramiento de calidad de la realización
Pruebas y evaluación	Administración de las evaluaciones y ajustes	Pruebas	Aseguramiento de calidad de las pruebas y evaluaciones
Cierre	Registro, publicación y entrega	Cierre	Entrega de software
Retroalimentación y mantenimiento	Seguimiento post-entrega	Mantenimiento	Seguimiento y propuestas de entrega

Tabla 2.1 Tareas principales de cada área en cada fase.

La figura 2.1 muestra el proceso utilizado por el grupo ESIE para desarrollar software multimedia interactivo, siguiendo el modelo de Boehm presentado en la sección anterior.

De manera general, la iteración más interna de la espiral es el inicio del proceso, comienza en la fase de planteamiento y análisis en el área administrativa con la creación del guion conceptual en el cual queda asentado la temática de la aplicación. Luego de la revisión del cliente de este guion, en las fases de diseño y realización el área de desarrollo se encarga de la creación de prototipos que son presentados al cliente para terminar de definir las características y el comportamiento del software. Cabe señalar que el área de control de calidad deberá asegurar la calidad de los productos de cada iteración. El proceso termina hasta que todas las áreas hayan ejecutado las tareas principales descritas en la tabla 2.1.

Más adelante se detallan las tareas y productos desarrollados en cada fase.

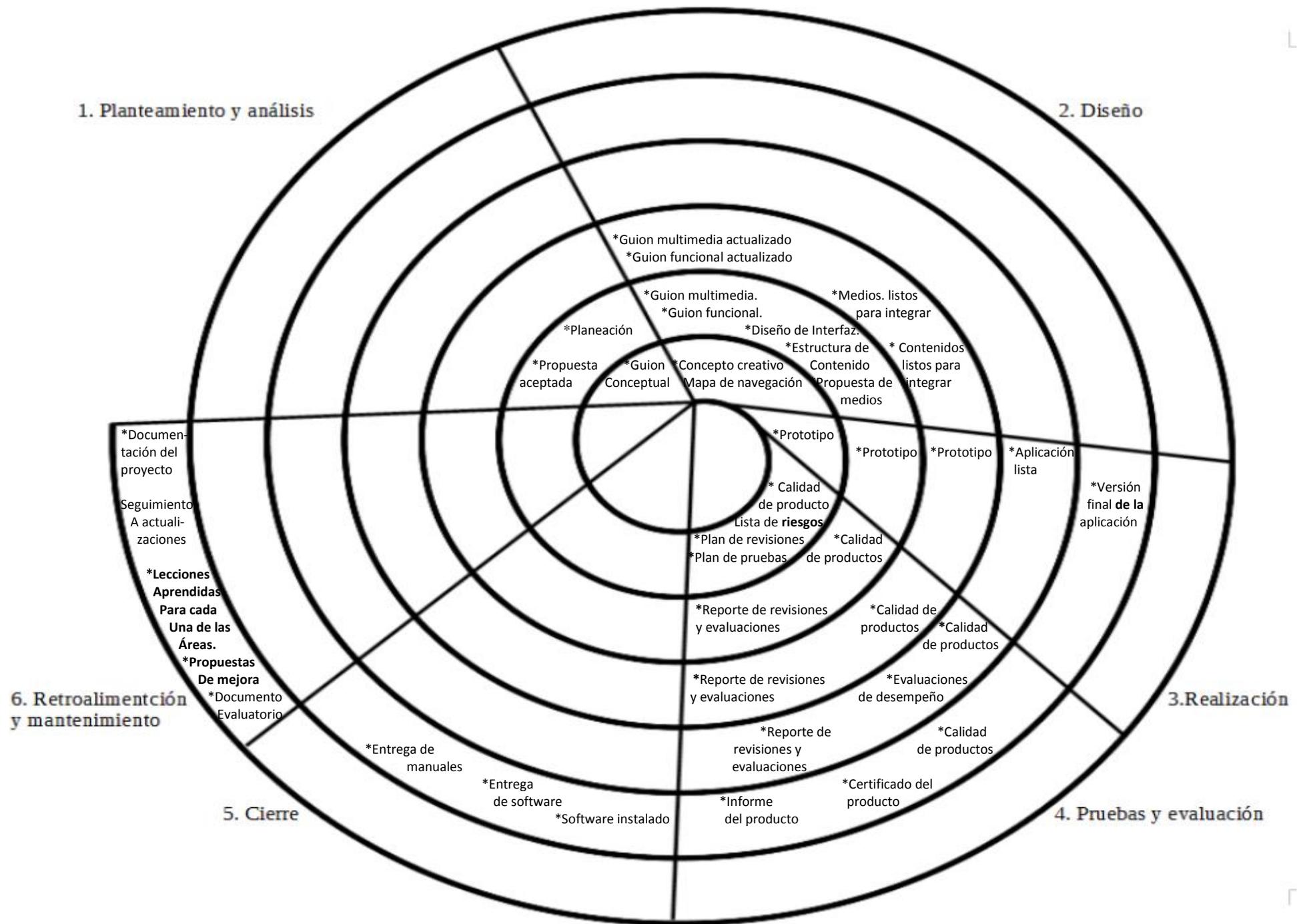


Figura 2.1 Proceso de software del grupo ESIE.

2.1 Roles

Durante este proceso intervienen los siguientes roles:

Cliente: Es la persona encargada de plantear la idea inicial del proyecto y la que solicita los servicios de una organización o institución para realizarlo, con el fin de venderlo o distribuirlo, además de fijar los objetivos y la fecha de entrega del proyecto, evalúa el costo, tiempo y riesgo que representa invertir en dicho producto.

Coordinador general: Es la persona que representa al grupo que desarrollará el software multimedia interactivo. Es el encargado de convencer al cliente de que su organización es la indicada para realizar el proyecto, es el responsable operativo y administrativo del equipo de trabajo. Debe de ser un especialista en el diseño de elementos multimedia y conocer a fondo los aspectos del proyecto para poder realizar un seguimiento puntual del mismo.

Administrador de proyecto: Es el encargado de elaborar la planeación del desarrollo del proyecto, establece junto con el coordinador general, los costos, da seguimiento a gastos, riesgos y además debe conocer todos los aspectos del producto a desarrollar para que pueda recopilar toda la información generada a lo largo del proyecto.

Experto en contenidos: Es la persona experta en la temática relacionada con el software y la que dará los contenidos necesarios para la aplicación.

Coordinador de contenidos: Recabará documentación y realizará entrevistas con las personas especializadas en la temática asociada al software para definir el contenido informativo que tendrá el producto.

Coordinador de desarrollo: Está a cargo del equipo de personas que implementarán el sistema en un lenguaje de programación, revisa diversas tecnologías viables para implementar la aplicación y además es el encargado de analizar los riesgos tecnológicos que pudiera llegar a tener el desarrollo de la aplicación.

Programador: Es el encargado de desarrollar el software que se ejecutará en una computadora y produce como resultado el funcionamiento esperado del sistema.

Guionista: Es el responsable de elaborar, con el apoyo de otros roles, los guiones que servirán de guía durante el proceso.

Líder de proyecto: Tendrá la responsabilidad de coordinar a los encargados de desarrollo, guiones y contenidos, debe convocar reuniones con los equipos de trabajo en donde realizará agendas y minutas con base en lo acordado.

Administrador de calidad: Es el encargado de definir los estándares para toda la documentación y llevar a cabo el control de las diferentes pruebas que se realicen a la aplicación. Además garantiza que el producto de software cumpla con todos los requerimientos solicitados.

Experto en evaluaciones: Es el responsable de diseñar y aplicar pruebas al software mediante evaluaciones con usuarios, este rol ayuda a garantizar la calidad del producto.

Diseñador gráfico: El diseñador se encarga de hacer las propuestas de diseño y estilo visual que tendrá la aplicación.

Experto en medios: Es la persona experta en medios multimedia, ya sea audio, vídeo o animación responsable de la creación de los medios que vayan acorde a los requerimientos del software.

Coordinador de medios: Tiene la responsabilidad de coordinar la creación de los medios (audio, video y animaciones) que se integrarán al software. Apoya la comunicación entre los guionistas y los creadores de los medios para que los productos se ajusten a las necesidades del producto.

Coordinador de interacción: Es la persona experta en la interacción humano-computadora, su responsabilidad principal es cuidar el diseño de la interfaz, propondrá ideas mediante bocetos hechos en papel sobre cómo debería ser la interacción con la aplicación.

En las secciones siguientes se describe las responsabilidades de las personas que tengan asignados estos roles en cada fase.

2.2 Planteamiento y análisis

En la fase de planteamiento y análisis se establece y analiza la idea del cliente, el equipo de trabajo deber de tener perfectamente claro la temática y alcances del proyecto a realizar, es necesario identificar a qué tipo de usuario irá dirigido el software, saber qué contexto de uso tendrá, además de contar con la colaboración de personas expertas en el tema de la aplicación.

Se formalizará, a través de un contrato, la responsabilidad que la organización adquiere con el cliente para realizar el software multimedia interactivo a través de un acuerdo económico. Por último el área de calidad valida estos productos, realiza un plan de riesgos para evitar retrasos en entregas y crea su plan de revisiones y plan de pruebas.

Áreas	Administrativa		Desarrollo		Control de Calidad
	Equipos	Coordinación general	Coordinación de proyecto	Creativo	Programación
Roles	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Cliente 	<ul style="list-style-type: none"> ✓ Administrador de proyecto ✓ Líder de proyecto 	<ul style="list-style-type: none"> ✓ Coordinador de medios. ✓ Coordinador de interacción. ✓ Guionista ✓ Coordinador de contenidos ✓ Líder de proyecto ✓ Coordinador de desarrollo. 	<ul style="list-style-type: none"> ✓ Diseñador gráfico. ✓ Programador. ✓ Experto en contenidos. ✓ Experto en medios. 	<ul style="list-style-type: none"> ✓ Administrador de calidad ✓ Administrador de proyecto
Tareas	<ul style="list-style-type: none"> • Recibir propuesta del cliente. • Transmitir idea al equipo de trabajo. • Entregar propuesta inicial. • Guion conceptual. • Estilo gráfico. • Prototipo esquemático. • Elaboración de costos. • Calendario preliminar. • Enfoque de contenido. • Gestionar aceptación de propuesta. 	<ul style="list-style-type: none"> • Coordinar elaboración del concepto. • Coordinar elaboración de guion conceptual. • Realizar el análisis de factibilidad. • Elaboración de costos. • Calendario preliminar. 	<ul style="list-style-type: none"> • Revisar software propio y ajeno. • Desarrollar concepto visual y/o sonoro. • Propuesta de plataforma tecnológica. • Diseño general de la interacción con el software. 	<ul style="list-style-type: none"> • Análisis de diseño y estilos gráficos. • Elaboración de mapa de navegación y prototipo interactivo. • Análisis de contenidos. 	<ul style="list-style-type: none"> • Lista de riesgos. • Plan de revisiones. • Plan de pruebas. • Plantillas de documentación.
Productos	<ul style="list-style-type: none"> • Documento de comentarios a la propuesta. • Documento de aceptación de propuesta (Contrato, convenio, carta de aceptación de la propuesta). 	<ul style="list-style-type: none"> • Guion conceptual. • Costos. • Calendario preliminar. • Propuesta inicial. • Formar carpeta de entrega. 	<ul style="list-style-type: none"> • Concepto creativo 	<ul style="list-style-type: none"> • Estilo gráfico. • Estilo sonoro. • Mapa de navegación. • Prototipo esquemático. • Enfoque de contenidos. 	<ul style="list-style-type: none"> • Lista de riesgos. • Plan de revisiones. • Plan de pruebas.

Los roles, tareas y productos generados se presentan en la tabla 2.2.

2.3 Diseño

Mediante propuestas de diseño, medios, contenidos e interacción se diseña un prototipo esquemático que sea capaz de representar la idea planteada en el guion conceptual, para así poder presentarlo al cliente y establecer los requerimientos necesarios del software. Se realiza un cronograma de actividades en el cual se asignan tareas, responsables y fechas de entrega, además de crear los guiones multimedia y funcional que son documentos de suma importancia ya que en ellos se describe a detalle diseño y funcionalidad del software.

El área de calidad con base en su plan de evaluaciones y pruebas genera un reporte de revisiones y evaluaciones de los productos generados.

Áreas	Administrativa		Desarrollo				Control de Calidad
	Equipos	Coordinación general	Coordinación de proyecto	Medios	Programación	Guiones	Contenidos
Roles	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Líder de Proyecto ✓ Administrador de proyecto 	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Líder de Proyecto ✓ Administrador de proyecto ✓ Coordinador de medios, interacción y contenidos ✓ Guionista ✓ Administrador de calidad 	<ul style="list-style-type: none"> ✓ Líder de proyecto ✓ Coordinador de medios ✓ Coordinador de interacción ✓ Diseñador gráfico ✓ Experto en medios 	<ul style="list-style-type: none"> ✓ Líder de proyecto ✓ Coordinador de desarrollo ✓ Programador 	<ul style="list-style-type: none"> ✓ Líder de desarrollo ✓ Líder de proyecto ✓ Guionista 	<ul style="list-style-type: none"> ✓ Líder de proyecto ✓ Experto en contenidos 	<ul style="list-style-type: none"> ✓ Administrador de calidad ✓ Administrador de proyecto
Tareas	<ul style="list-style-type: none"> • Entregar propuesta de diseño: Bocetos y prototipo esquemático del interactivo. • Estructura de los contenidos. • Gestionar aceptación de propuesta. 	<ul style="list-style-type: none"> • Coordinar elaboración del concepto. • Coordinar elaboración de guion conceptual. • Realizar el análisis de factibilidad. • Elaboración de costos. • Cronograma de actividades preliminar. 	<ul style="list-style-type: none"> • Asignar tareas según cronograma de actividades. • Análisis de medios. • Análisis de estilo gráfico. • Diseño de interacción. • Análisis de audio o video. 	<ul style="list-style-type: none"> • Asignar tareas según cronograma de actividades. • Arquitectura de software. • Prototipo esquemático de la aplicación. • Diseño de interacción. • Plan de versiones. 	<ul style="list-style-type: none"> • Asignar tareas según cronograma de actividades. • Realizar guion multimedia. • Realizar guion funcional. 	<ul style="list-style-type: none"> • Asignar tareas según cronograma de actividades. • Realizar propuesta de la estructura de contenidos. 	<ul style="list-style-type: none"> • Lista de riesgos. • Plan de revisiones. • Plan de pruebas. • Plantillas de documentación.
Productos	<ul style="list-style-type: none"> • Documento de comentarios a la propuesta. • Documento de aceptación de propuesta. 	<ul style="list-style-type: none"> • Cronograma de actividades. • Equipo de desarrollo conformado. • Plan de adquisición de material para desarrollo. 	<ul style="list-style-type: none"> • Seguimiento de actividades. • Propuesta de medios. • Propuesta de estilo gráfico. 	<ul style="list-style-type: none"> • Seguimiento de actividades. • Arquitectura del software. • Prototipo esquemático interactivo. 	<ul style="list-style-type: none"> • Seguimiento al cronograma de actividades. • Guion multimedia. • Guion funcional. 	<ul style="list-style-type: none"> • Seguimiento de actividades. • Estructura de los contenidos. 	<ul style="list-style-type: none"> • Reporte de revisiones y evaluaciones.

Los roles, tareas y productos generados se presentan en la tabla 2.3

2.4 Realización

Esta fase estará centrada principalmente en realizar la interfaz gráfica, contenidos preliminares y medios (audio, vídeo o animaciones) que sean necesarios para después integrarlos para cumplir la funcionalidad requerida del software. Esto dará como resultado un prototipo que quedará listo para evaluarse. Estas tareas recaen principalmente en el área de desarrollo.

Se actualizarán los guiones multimedia y funcional con base en los nuevos requerimientos del software y el área de calidad generará reportes de evaluaciones y pruebas hechas al prototipo.

Áreas	Administrativa		Desarrollo				Control de Calidad
	Equipos	Coordinación general	Coordinación de proyecto	Medios	Programación	Guiones	Contenidos
Roles	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Líder de Proyecto ✓ Administrador de proyecto 	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Líder proyecto ✓ Administrador de proyecto ✓ Coordinador de medios, interacción y contenidos ✓ Guionista ✓ Administrador de calidad 	<ul style="list-style-type: none"> ✓ Coordinador de medios ✓ Líder de proyecto ✓ Diseñador gráfico ✓ Experto en medios 	<ul style="list-style-type: none"> ✓ Coordinador de desarrollo ✓ Programador 	<ul style="list-style-type: none"> ✓ Guionista ✓ Coordinador de interacción 	<ul style="list-style-type: none"> ✓ Experto en contenidos ✓ Coordinador de contenidos 	<ul style="list-style-type: none"> ✓ Administrador de calidad ✓ Administrador de proyecto
Tareas	<ul style="list-style-type: none"> • Seguimiento al cronograma de actividades. • Gestionar requerimientos legales y logísticos de la producción. 	<ul style="list-style-type: none"> • Seguimiento al cronograma de actividades. 	<ul style="list-style-type: none"> • Asignar tareas según cronograma de actividades • Seguimiento de las tareas • Realizar interfaz gráfica • Realizar medios 	<ul style="list-style-type: none"> • Asignar tareas según cronograma de actividades. • Seguimiento de las tareas. • Integrar y programar la aplicación. • Dar seguimiento de avances y pruebas unitarias. 	<ul style="list-style-type: none"> • Realizar modificaciones al guion multimedia en caso de ser necesario. • Realizar modificaciones al guion funcional en caso de ser necesario. 	<ul style="list-style-type: none"> • Desarrollo de contenidos. 	<ul style="list-style-type: none"> • Lista de riesgos. • Plan de revisiones. • Plan de pruebas.
Productos	<ul style="list-style-type: none"> • Permisos de uso y reproducción. 	<ul style="list-style-type: none"> • Lista de ajustes a planes 	<ul style="list-style-type: none"> • Medios listos para integrar (audio, video o animaciones) • Elementos gráficos listos para integrar 	<ul style="list-style-type: none"> • Prototipo listo para evaluar. 	<ul style="list-style-type: none"> • Seguimiento de actividades. • Guion multimedia actualizado. • Guion funcional actualizado. 	<ul style="list-style-type: none"> • Contenidos listos para integrar • Guion de contenidos 	<ul style="list-style-type: none"> • Reporte de revisiones y evaluaciones.

Los roles, tareas y productos generados se presentan en la tabla 2.4

2.5 Pruebas y evaluación

Una vez que el prototipo ha sido desarrollado se hará la integración de contenidos finales, los expertos en evaluaciones lo pondrán a prueba para observar su funcionamiento. Mediante evaluaciones con usuarios reales, se hará uso del software tal cual si estuviese en un ambiente real y productivo se tratará de obtener la percepción que tiene el usuario hacia él para detectar fallos y evaluar si el producto está cumpliendo el propósito por el cual fue desarrollado. Con base en la retroalimentación obtenida de las pruebas se corrige el software para tener incrementos y mejoras tanto de calidad y funcionalidad.

El área de calidad realizará evaluaciones de desempeño y validará que las correcciones surgidas en las pruebas se realicen en tiempo y forma.

Áreas	Administrativa		Desarrollo			Control de Calidad	
	Equipos	Coordinación general	Coordinación de proyecto	Medios	Programación	Guiones	Calidad
Roles	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Líder de Proyecto ✓ Administrador de proyecto 	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Líder de proyecto ✓ Administrador de proyecto ✓ Coordinador de medios, interacción y contenidos. ✓ Guionista ✓ Administrador de calidad 	<ul style="list-style-type: none"> ✓ Coordinador de medios ✓ Coordinador de interacción ✓ Líder de proyecto ✓ Diseñador gráfico ✓ Experto en medios 	<ul style="list-style-type: none"> ✓ Coordinador de desarrollo ✓ Programador 	<ul style="list-style-type: none"> ✓ Guionista 	<ul style="list-style-type: none"> ✓ Administrador de calidad ✓ Administrador de proyecto 	<ul style="list-style-type: none"> ✓ Equipo de expertos en evaluaciones con usuarios
Tareas	<ul style="list-style-type: none"> • Presentar plan de evaluaciones. • Presentar reporte de evaluaciones al cliente. • Presentar plan de ajustes según resultados. 	<ul style="list-style-type: none"> • Ajustar calendario y tiempos • Gestionar la realización de las evaluaciones con usuario y cliente. 	<ul style="list-style-type: none"> • Corregir problemas encontrados derivados de las evaluaciones. • Entrega final de medios. • Integración final. 	<ul style="list-style-type: none"> • Realizar pruebas de funcionamiento y plataformas. • Corregir problemas encontrados derivados de las evaluaciones. • Integración final. 	<ul style="list-style-type: none"> • Corregir problemas encontrados derivados de las evaluaciones. • Entrega final de contenidos. 	<ul style="list-style-type: none"> • Gestionar la realización de las evaluaciones con usuarios. • Canalizar correcciones derivadas de la evaluación al área correspondiente. 	<ul style="list-style-type: none"> • Realizar evaluaciones con usuarios. • Realizar análisis de evaluaciones. • Entregar reporte de evaluaciones.
Productos	<ul style="list-style-type: none"> • Carta de conformidad con cambios propuestos. 	<ul style="list-style-type: none"> • Plan de evaluaciones • Plan de entregas ajustado • Plan de costos ajustado 	<ul style="list-style-type: none"> • Versión beta del software. 			<ul style="list-style-type: none"> • Verificar que las correcciones se hagan en tiempo y forma. • Evaluaciones de desempeño. 	<ul style="list-style-type: none"> • Evaluaciones con usuarios. • Reporte de evaluaciones.

Los roles, tareas y productos generados se presentan en la tabla 2.5

2.6 Cierre

Se concluye el desarrollo para generar una versión final del software y se procede a realizar actividades finales como el modo de distribución, el diseño de la presentación visual del disco de instalación, en caso de ser necesaria, manuales de uso e instalación del software. Se genera un reporte de conclusión, en el cual queda establecido que todos los productos de entrega están cumpliendo con la calidad y los estándares propuestos.

Áreas	Administrativa		Desarrollo			Control de Calidad
Equipos	Coordinación general	Coordinación de proyecto	Medios	Programación	Guiones	Calidad
Roles	<ul style="list-style-type: none"> ✓ Coordinador general ✓ Líder de proyecto ✓ Administrador de proyecto 	<ul style="list-style-type: none"> ✓ Coordinador general. ✓ Líder de proyecto ✓ Administrador de proyecto ✓ Administrador de calidad ✓ Experto en contenidos ✓ Coordinador de contenidos 	<ul style="list-style-type: none"> ✓ Coordinador de medios ✓ Coordinador de interacción ✓ Líder de proyecto ✓ Diseñador gráfico ✓ Experto en audio 	<ul style="list-style-type: none"> ✓ Coordinador de desarrollo ✓ Programador 	<ul style="list-style-type: none"> ✓ Guionista 	<ul style="list-style-type: none"> ✓ Administrador de calidad ✓ Administrador de proyecto
Tareas	<ul style="list-style-type: none"> • Presentación final 	<ul style="list-style-type: none"> • Preparación final de entrega. • Creación del reporte técnico 	<ul style="list-style-type: none"> • Compilación de versión final del software. • Creación de manuales de uso e instalación. 			<ul style="list-style-type: none"> • Revisar que los documentos de entrega cumplen los estándares propuestos.
Productos	<ul style="list-style-type: none"> • Entrega final. 	<ul style="list-style-type: none"> • Entrega final de software y manuales. • Reporte técnico 	<ul style="list-style-type: none"> • Versión final del software. • Manuales de uso e instalación. 			<ul style="list-style-type: none"> • Reporte de verificación del producto.

Los roles, tareas y productos generados se presentan en la tabla 2.6

2.7 Retroalimentación y mantenimiento

Una vez que la aplicación está en un ambiente productivo se monitorea el funcionamiento y se documenta su uso para realizar mejoras a la funcionalidad, en caso de ser necesario, a su vez cada área se encarga de generar un documento con las lecciones aprendidas durante el desarrollo del software para detectar fallos y enriquecer el proceso.

Áreas	Administrativa	Desarrollo	Control de Calidad
Equipos	Coordinación general	Coordinación de proyecto	Calidad
Roles	✓ Administrador de proyecto	✓ Coordinador de desarrollo ✓ Programador	✓ Administrador de calidad ✓ Administrador de proyecto
Tareas	<ul style="list-style-type: none"> Realizar trabajo de campo. Generar estadísticas de evaluación y retroalimentación. Registrar lecciones aprendidas. Terminar de integrar la documentación del proyecto. 	<ul style="list-style-type: none"> Registrar lecciones aprendidas sobre. <ul style="list-style-type: none"> Proceso de desarrollo Equipo de desarrollo Producto Certificar el producto. Actualizar contenidos, funcionalidad y plataformas. 	<ul style="list-style-type: none"> Realizar actualizaciones en los estándares que se aplican al producto. Dar seguimiento a la consistencia en las actualizaciones. Hacer propuestas de mejoras en cuanto a la calidad del proceso y del producto. Registrar lecciones aprendidas.
Productos	<ul style="list-style-type: none"> Lecciones aprendidas de la administración del proyecto. Documentación del proyecto. 	<ul style="list-style-type: none"> Lecciones aprendidas del proceso de desarrollo. Software actualizado. 	<ul style="list-style-type: none"> Lecciones aprendidas del aseguramiento de calidad. Propuestas de mejora. Seguimiento a las actualizaciones.

Los roles, tareas y productos generados se presentan en la tabla 2.7

2.8 Resumen

El software que use medios de comunicación tales como texto, imagen, sonido, animación o video de una forma combinada y congruente para cumplir los objetivos por el cual se desea desarrollar, se denomina software multimedia interactivo. Este capítulo describe un proceso iterativo que permite el desarrollo de este tipo de software, se compone por seis fases: Planteamiento y análisis, diseño, realización, pruebas y evaluación, cierre y como última fase retroalimentación y mantenimiento.

El proceso reparte las tareas de las fases en tres áreas: Administrativa, desarrollo y control de calidad, cada área tiene una tarea principal que cumplir en cada fase, las áreas cuentan con equipos de trabajo en donde cada persona tendrá uno o más roles con ciertas responsabilidades.

Con base a esto se pretende dejar en claro que un proceso debe definir fases, roles, tareas y productos.

El siguiente capítulo mostrará cómo se aplicó este proceso en el desarrollo del software interactivo *Explora la función.*

3 Aplicación del proceso de desarrollo de software multimedia interactivo

El proceso de software descrito al final del capítulo anterior, ha sido utilizado por el grupo ESIE para el desarrollo de distintos interactivos teniendo como resultado productos exitosos y de calidad.

Tomando en cuenta que es un proceso iterativo, el desarrollo de la aplicación tuvo varias iteraciones del proceso y mostrar el resultado de cada iteración haría que el contenido del presente trabajo se extendiera en demasía, con base en esto las siguientes secciones mostrarán en términos generales cómo fue el desarrollo del software multimedia interactivo *Explora la función* y el resultado final obtenido.

3.1 Planteamiento y análisis

La empresa “Sietecolores, Ideas Interactivas” solicitó al grupo ESIE el desarrollo de un interactivo llamado *Explora la función*.

Para establecer con claridad las características del software, fue necesario realizar una serie de reuniones entre el grupo ESIE y el cliente: “Sietecolores, Ideas Interactivas”. Como resultado de estas reuniones, se definieron las características de este software que se presentan en este capítulo.

3.1.1 Tema del interactivo

El interactivo *Explora la función* abordará el tema de la diversidad biológica que habita en el suelo de determinadas zonas de Nuevo León. Bajo este contexto, se considerará que el suelo es un hábitat que alberga una amplia gama de organismos dentro de los tres dominios taxonómicos. Cuando se hace el análisis del suelo, se estudia la contribución de los organismos en los diferentes procesos del ecosistema, por ejemplo, la descomposición de materia orgánica y su impacto en el desarrollo del ecosistema. Así, el interactivo deberá proporcionar información sobre el impacto que tienen las especies de la fauna edáfica en el desarrollo del ecosistema.

3.1.2 Objetivos

Los objetivos específicos que tiene este interactivo son los siguientes:

- Identificar al suelo como un microhábitat donde se pueden encontrar una gran diversidad de especies.
- Reconocer los cambios que ocurren en un microhábitat de acuerdo a las características del suelo y la fauna que lo habita.

3.1.3 Perfil de usuario

El software está dirigido principalmente para niños a partir de 7 años en adelante que habiten en el estado de Nuevo León. Se plantea que el interactivo será usado por un sólo usuario a la vez.

3.1.4 Contexto de uso

El interactivo formará parte de las exhibiciones del museo Papalote Verde ubicado en Monterrey, Nuevo León dentro del Parque Fundidora. El museo está enfocado a fomentar la cultura de educación ambiental en la región. El espacio museográfico se distribuye en cinco zonas temáticas permanentes: *Pertenezco*, *Comunicó*, *Comprendo*, *Expreso* y *Soy*, y una zona permanente para niños pequeños llamada *Bosque y Mar*.

La exhibición de *Explora la función* se encuentra en la zona de *Pertenezco*, y muestra a través de tres dispositivos, las interacciones e interrelaciones de los organismos que habitan e intervienen en los procesos del suelo en sus diferentes escalas: Microfauna, Mesofauna y Macrofauna.

3.1.5 Descripción del interactivo

Se desarrollaron tres interactivos, uno para cada escala de fauna, pero el diseño está basado en el mismo concepto. El presente trabajo se centrará específicamente en describir el desarrollo del software para el caso de Macrofauna. El interactivo propone que los participantes interactúen de manera individual con un dispositivo que permita la interacción multitáctil.

La propuesta plantea mostrar información de dos ecosistemas diferentes del estado de Nuevo León: bosque y matorral. Ver figura 3.1

Suelo y Biodiversidad Macrofauna



Figura 3.1 Selección de ecosistema.

Una vez que el usuario haya seleccionado un tipo de ecosistema en la figura 3.1, el interactivo permitirá agregar organismos de ciertas especies al ambiente y observar cómo cambia el entorno con su presencia, para ello se debe especificar el tiempo y las condiciones climáticas que se simularán, las cuales pueden ser condiciones favorables o desfavorables para la especie (Ver figura 3.2).

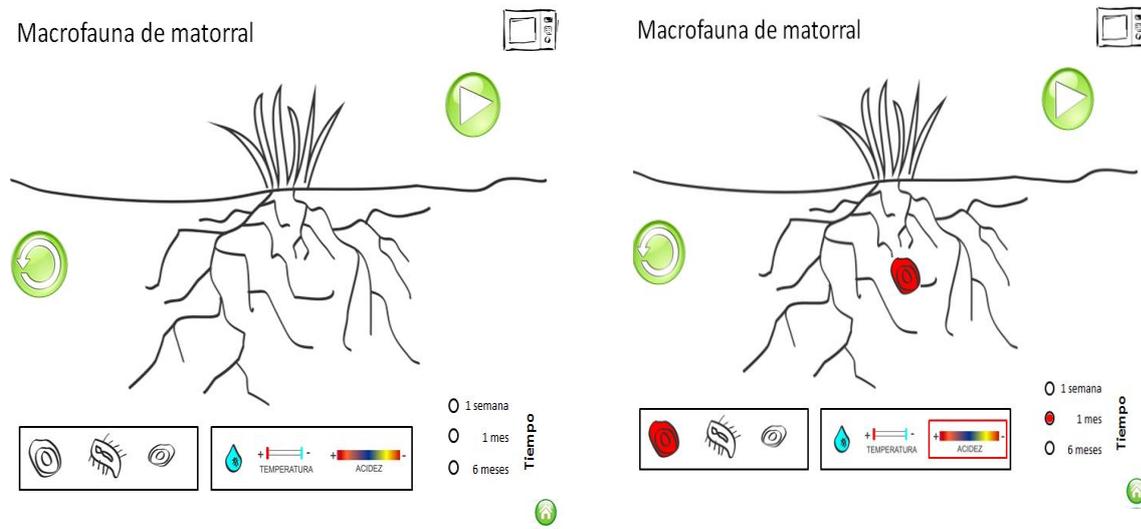


Figura 3.2 Agregar organismos al ambiente.

Basándose en la especie elegida y los parámetros asignados se realizarán simulaciones que permitan observar el impacto que se provoca en el ambiente. Ver figura 3.3

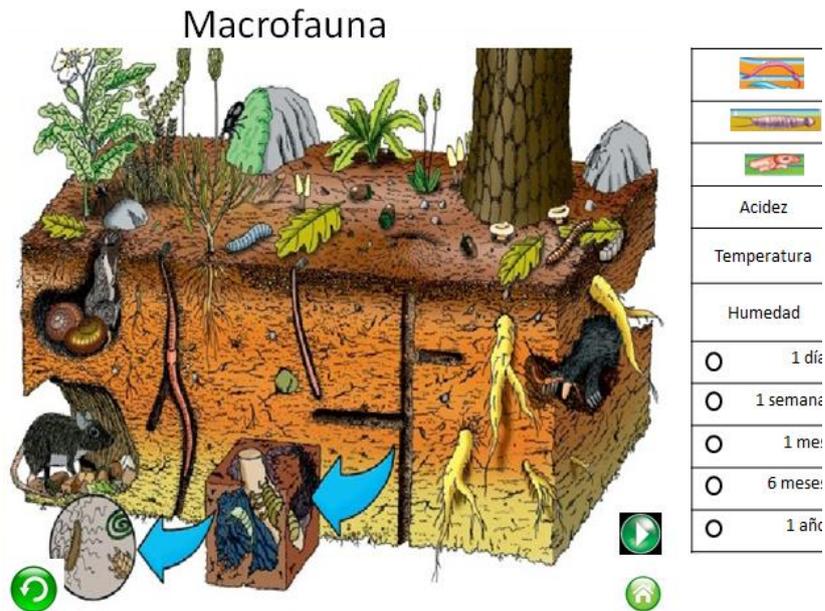


Figura 3.3 Simulación de cambios de ambiente.

3.1.5.1 Especificaciones de Hardware

Los requerimientos de hardware establecidos por el cliente para el funcionamiento del interactivo son una tableta electrónica Surface (ver figura 3.4) para cada escala de fauna, con las siguientes características:

- Windows 8 Professional
- Memoria RAM de 4GB
- Pantalla 10.6" ClearType Full HD
- Resolución 1920x1080 pixels de 16:9 (widescreen)
- Multi-toque de 10 puntos
- Procesador Intel Core i5 3ra Generación
- Tarjeta de video Intel HD Graphics 4000
- Tarjeta de red Wi-Fi (802.11a/b/g/n)



Figura 3.4 Tableta Surface.

3.1.5.2 Especificaciones de software

Dado el perfil de usuario que tiene el interactivo se deseaba que fuera visualmente atractivo, tuviera imágenes en alta resolución, contará con gráficos vectoriales, y animación digital. *Explora la Función* se realizó usando tres aplicaciones de Adobe System Incorporated, este tipo de software está enfocado en aspectos multimedia.

Para la cuestión de imágenes se utilizó Adobe Photoshop CS6 y Adobe Illustrator CS6, mientras que para animaciones y funcionalidad del interactivo se usó Adobe Flash Professional CS6. Los proyectos de Flash abarcan desde simples animaciones hasta el desarrollo de software complejo como juegos, interfaces de usuario, aplicaciones de internet e interactivos como *Explora la Función*, que cuentan con una amplia variedad de contenido multimedia. Flash permite desarrollar la funcionalidad del interactivo en un lenguaje de programación llamado ActionScript 3.0, es un lenguaje robusto, usa el paradigma de orientación a objetos, permite definir código en una serie de scripts y crear objetos llamados **símbolos**¹. Para poder ejecutar la aplicación será necesario instalar Adobe Flash Player en la tableta Surface. Esta información queda asentada en el guion conceptual como principal producto generado de esta fase.

3.2 Diseño

Una vez que “Sietecolores, Ideas Interactivas” aceptó la propuesta general del interactivo, descrita en el guion conceptual, se comenzó con el diseño de software. Como un primer resultado, el área de desarrollo realizó propuestas de diseño, interacción y funcionalidad. Se hicieron bocetos en papel y mapas de navegación para armar prototipos y definir con detalle el funcionamiento del software para revisarlo con el cliente. Al final se propuso una interacción simple basada en gestos sobre la tableta, que busca la integración de los objetivos de aprendizaje de manera natural.

Además de plantear la funcionalidad del interactivo, el grupo ESIE también fue el responsable de los contenidos didácticos. Con la ayuda del experto en contenidos se realizó una investigación sobre algunos de los organismos más representativos que habitan cada ecosistema. Se seleccionó un conjunto de ellos y se presentaron a “Sietecolores, Ideas Interactivas” para que aprobará las especies que aparecen en el interactivo. Para cada escala de fauna se crearon seis organismos, tres para bosque y tres para matorral.

A continuación se muestra la comparación de diseño de pantallas del prototipo esquemático hecho en la fase uno, con algunas pantallas del diseño final y además se describe el funcionamiento de *Explora la función*.

¹ Son elementos autónomos que se pueden crear y manipular de una manera específica dentro de Flash, un símbolo puede ser un Gráfico, un Botón o un Clip de película. Un símbolo de tipo clip de película puede encapsular gráficos importados de Photoshop o Illustrator, botones, incluso otros clips de película.

3.2.1 Diseño de propuestas gráficas

Una vez aprobado el estilo gráfico para el interactivo, se diseñaron las diferentes pantallas tomando como referencia el funcionamiento propuesto en el prototipo. En la figura 3.5 se muestra la comparación de la pantalla de selección del ecosistema a la izquierda la imagen del prototipo y a la derecha la versión final. Se puede observar que ambas permiten la selección de un ecosistema y el inicio de la actividad, mientras que la versión final muestra la descripción y localización de un ecosistema a la vez para no saturar la pantalla con información.



Figura 3.5 Pantallas para descripción de ecosistema.

La figura 3.6 muestra una comparación entre la pantalla de trabajo del prototipo y la versión final, la imagen de la izquierda muestra un tipo de vegetación de matorral donde se puede agregar organismos que sean parte de ese hábitat, además se puede observar los botones de interacción: inicio, volver a pantalla inicial, reiniciar y los controles de tiempo. A la derecha observamos la versión final con un acercamiento a una planta propia del ecosistema matorral, los mismos botones que se presentan en el prototipo y el organismo representante de ese hábitat. Se puede observar que en las dos versiones se mantienen los controles de interacción y la idea original de agregar organismos arrastrándolos al ambiente como fue propuesto en la fase de planteamiento y análisis.

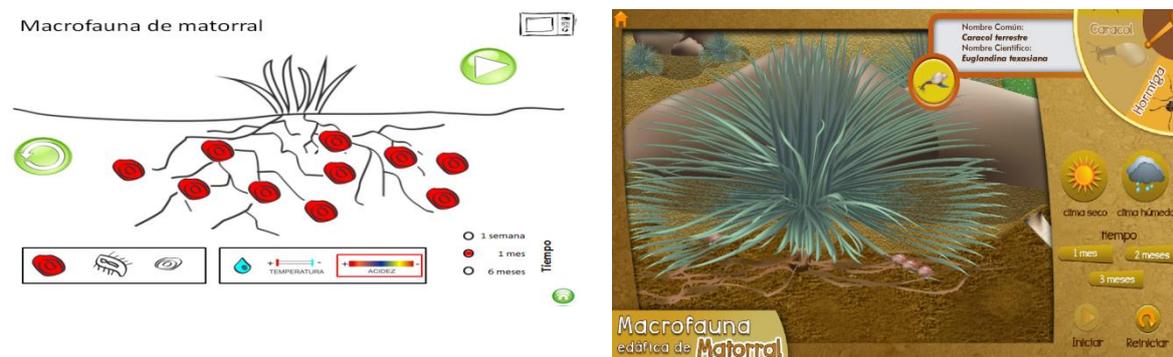


Figura 3.6 Pantallas de ambiente para agregar especies al ecosistema.

La figura 3.7 es una comparación de simulación específica para la lombriz del ecosistema bosque, la imagen de la izquierda muestra la versión del prototipo y a la derecha se muestra la versión final, en ambas propuestas el hábitat del organismo es el suelo, dependiendo de los

parámetros de tiempo y clima se observará que la población del organismo aumenta o disminuye.



Figura 3.7 Pantallas de simulaciones en los horizontes de suelo.

3.2.2 Estilo gráfico

Dado que el interactivo está pensado para que gran parte de sus usuarios sean niños, se hizo mucho énfasis en el diseño de organismos, flora y **horizontes**² para que fueran realistas, por lo que se debía tomar como referencia imágenes reales de los organismos así como horizontes del suelo y flora de ese tipo de ecosistemas.

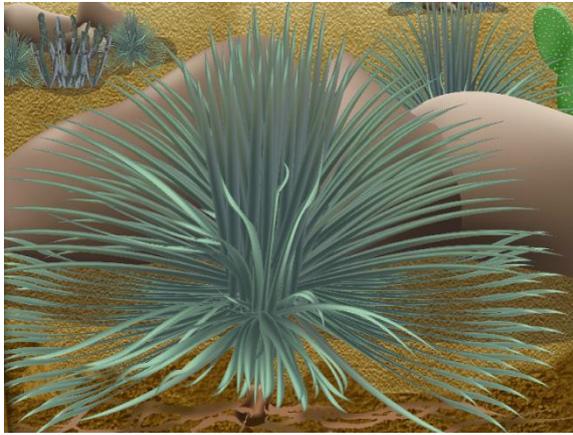
La figura 3.8 muestra una comparativa entre algunos organismos reales con sus diseños finales, se puede observar el trabajo de ilustración digital para lograr una semejanza entre la versión usada en el interactivo y la fotografía. Cabe mencionar que para lograr este tipo de ilustraciones fue indispensable la colaboración del experto de contenidos en la revisión de las propuestas.



Figura 3.8 Comparación visual de organismos.

² A cada capa en que se organiza el material del suelo se le denomina horizonte, son capas horizontales que se pueden diferenciar con base en su color, textura o estructura.

La flora de los ecosistemas fue parte importante al momento de diseñar los respectivos hábitats de las especies, la figura 3.9 muestra la comparación de una planta propia del ecosistema matorral, la imagen de la derecha es una de varias imágenes que se tomaron como referencia para realizar la versión digital de la izquierda, se puede observar que el diseño final conserva la forma y color respectivo de la original.



<http://davesgarden.com/members/Cactusjordi>

Figura 3.9 Comparación visual de flora matorral.

La comparación de horizontes de bosque se muestra en la figura 3.10, del lado izquierdo se muestra la versión final de diseño y a la derecha una de las imágenes que fueron tomadas como base para realizar la versión digital. En ambas imágenes se aprecia el cambio de color como un indicador de las capas que componen la estructura del suelo. Además de la similitud de la corteza y la profundidad de las raíces del árbol dentro del suelo.



Figura 3.10 Comparación visual de horizontes.

3.2.3 Descripción del funcionamiento

Cuando el interactivo no detecta actividad por tres minutos, se muestra una serie de animaciones de los diferentes organismos elegidos de cada tipo de fauna. La figura 3.11 muestra los organismos seleccionados para la macrofauna edáfica.



Figura 3.11 Animaciones de organismos.

Cuando el usuario toca la pantalla se detiene la animación y se muestra la pantalla inicial. La pantalla inicial (figura 3.12) mostrará las opciones de los ecosistemas con las animaciones de dos organismos, uno para cada ecosistema.



Figura 3.12 Pantalla Inicio Explora la función español.

Para iniciar la actividad el usuario debe tocar las áreas 1 o 2 de la pantalla para seleccionar el ecosistema. También se puede cambiar el idioma en el que se muestra la información en el

interactivo, el idioma predeterminado es español pero se puede cambiar con las opciones del área 3. La figura 3.13 muestra la pantalla inicial en inglés.



Figura 3.13 Pantalla Inicio Explora la función inglés.

Una vez elegido el tipo de ecosistema, se muestra un mapa de Nuevo León que presenta las zonas en donde existe el ecosistema elegido, una descripción del ecosistema y las animaciones de las especies que están presentes en el ecosistema. La figura 3.14 muestra las pantallas para la descripción del ecosistema bosque y matorral respectivamente.



Figura 3.14 Pantallas descripción de ecosistema.

Para empezar la actividad el usuario deberá de tocar el botón 5 o si lo prefiere podrá cambiar de ecosistema tocando el botón 4 (ver figura 3.15).



Figura 3.15 Pantalla descripción de bosque.

El funcionamiento para la pantalla de Matorral es análogo.

Una vez que el usuario seleccionó un tipo de fauna se mostrará la pantalla de instrucciones que incluye un video que muestra cómo usar el interactivo (ver figura 3.16). Al tocar el botón 6 la pantalla de instrucciones se cierra y se presenta la pantalla de trabajo.

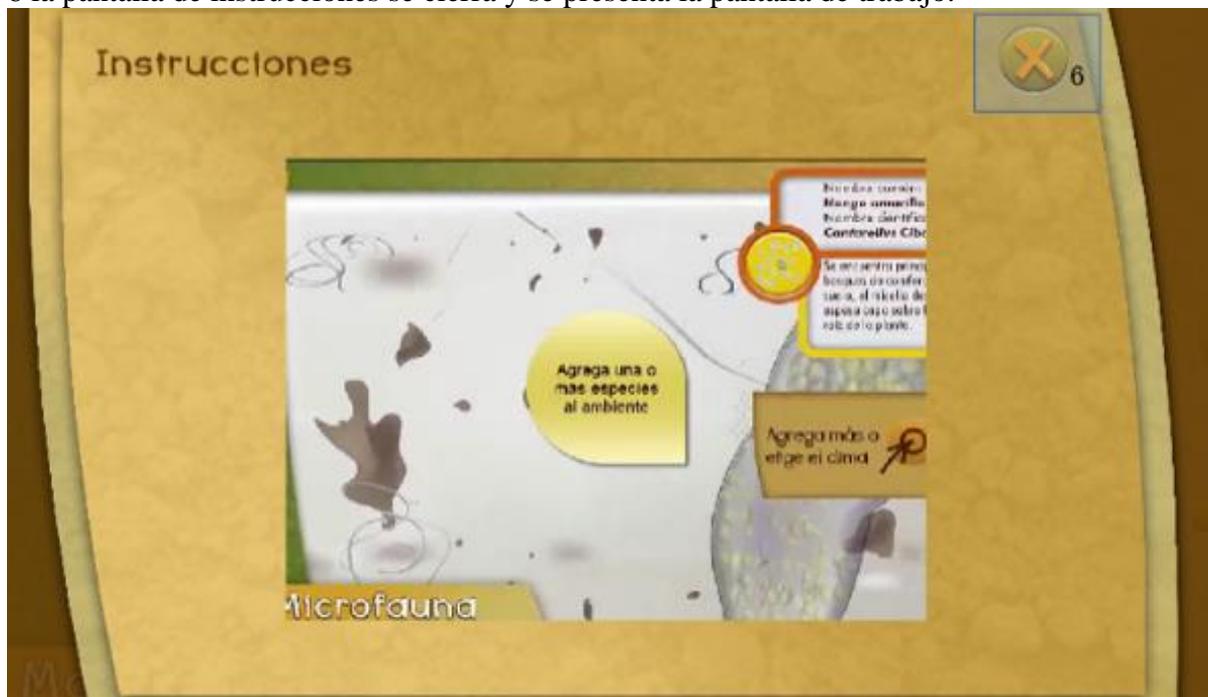


Figura 3.16 Pantalla de explicación final.

La figura 3.17 contiene los siguientes elementos y representa la pantalla de trabajo **Matorral**³.



Figura 3.17 Pantalla de trabajo Matorral.

A continuación se describen los elementos de la pantalla de trabajo:

- Título (7). Ubicado en el lado inferior izquierdo muestra el nombre del tipo de escala y ecosistema.
- Horizonte del ecosistema (8). Es el área principal del ecosistema ubicada en el centro de la pantalla, dependiendo de la elección del usuario se le realizará el zoom al hábitat de la especie.
- Ruleta animada con tres especies (9). Se encuentra en la parte superior derecha de la pantalla, tiene un ligero movimiento para que el usuario sepa que la puede girar, y contiene un indicador, el cual tiene como función mostrar los diferentes organismos que el usuario podrá utilizar para realizar el experimento. Aprovechando la tecnología táctil del dispositivo la ruleta girará dependiendo de la dirección que el usuario le de al desplazarla con un toque hacia arriba o hacia abajo. Cuando esto suceda aparecerá el organismo interactivo, la ficha informativa de este y se realizará el zoom respectivo a la zona del hábitat de la especie.
- Botones de clima (10). Ubicados en la parte inferior de la ruleta, los botones de clima tienen como función permitir al usuario seleccionar un clima seco o húmedo para realizar el experimento.
- Botones de tiempo (11). Se encuentran de lado inferior derecho y permiten elegir 1 mes, 2 meses o 3 meses respectivamente que es el tiempo que se va a simular que avanzará en el ecosistema.

³ El funcionamiento para la pantalla bosque es análogo.

- Botón iniciar (12). Se encuentra en la parte inferior derecha, este botón tiene como función dar inicio a la simulación del avance del tiempo y los cambios correspondientes en el ecosistema.
- Botón reiniciar (13). Se encuentra a un costado del botón iniciar, cuando el usuario toca este botón reinicia nuevamente al estado inicial del ecosistema.
- Botón regresar a página de inicio (14). Se encuentra en la parte superior izquierda, cuando el usuario toca este botón regresa a la pantalla de selección de hábitat.

La figura 3.18 muestra la pantalla de trabajo para la hormiga escamolera. Para cada especie, se muestra un ejemplar interactivo (15) y una ficha informativa del organismo elegido (16). La ficha informativa muestra la siguiente información:

- 1.- Nombre común y científico del organismo.
- 2.- Características del organismo.
- 3.- Ecología de la especie del organismo.
- 4.- Clima del organismo
- 5.- Explicación final.

Para ayudar al usuario a realizar la actividad, el interactivo utiliza seis ventanas de ayuda (17) que van apareciendo conforme el usuario avanza en la actividad. Las ventanas de ayuda aparecerán solamente en la primera simulación que realice el usuario.

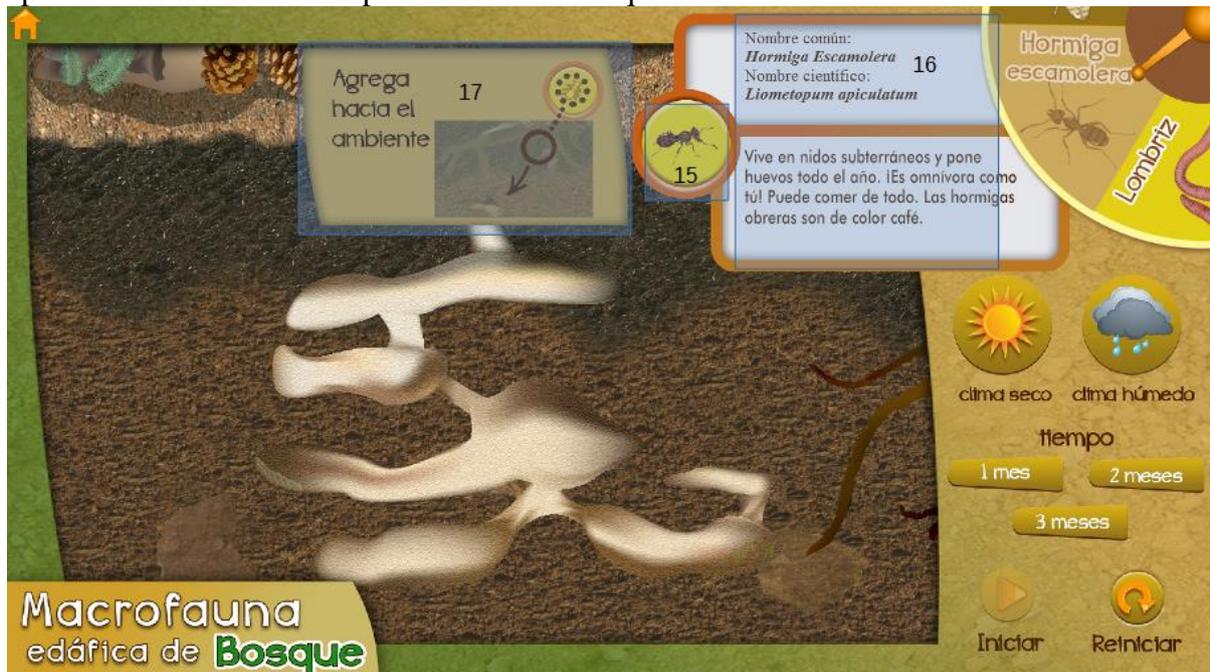


Figura 3.18 Acercamiento de horizonte.

Para realizar la actividad el usuario debe arrastrar al ejemplar de la especie a la zona del horizonte para que inicie la interacción con el microhábitat, al soltar al ejemplar se observa la animación de su comportamiento dentro del microhábitat. El usuario puede arrastrar cierto número de ejemplares al horizonte.

Posteriormente podrá modificar propiedades de clima, ya que la interacción de la especie se ve afectada por las condiciones generales del clima. También se debe especificar el tiempo que queremos que se simule para ver los efectos de la especie en el microhábitat.

Finalmente, al tocar el botón iniciar aparecerá una pantalla de tiempo emergente (ver figura 3.19) que representa mediante un reloj animado (18) el tiempo que ha pasado y muestra una ficha informativa (19) del impacto que tienen esos organismos en el ambiente. Al concluir la animación del reloj el usuario podrá tocar el botón (20) y la ventana emergente desaparecerá.

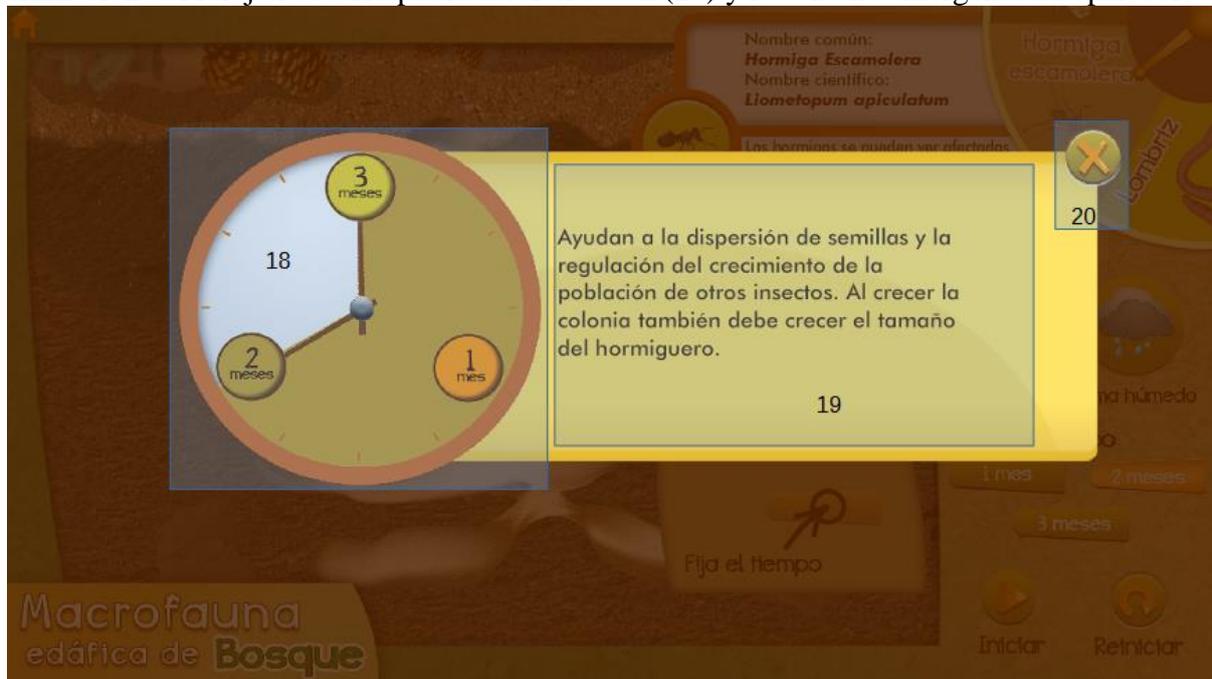


Figura 3.19 Pantalla simulación de tiempo.

Cuando la explicación anterior se oculta, el usuario podrá observar en el horizonte los posibles cambios provocados por la especie elegida. La figura 3.20, muestra los cambios que se pueden generar en el hábitat de la hormiga escamolera. El resto de simulaciones de especies serán mostradas en el Anexo A.

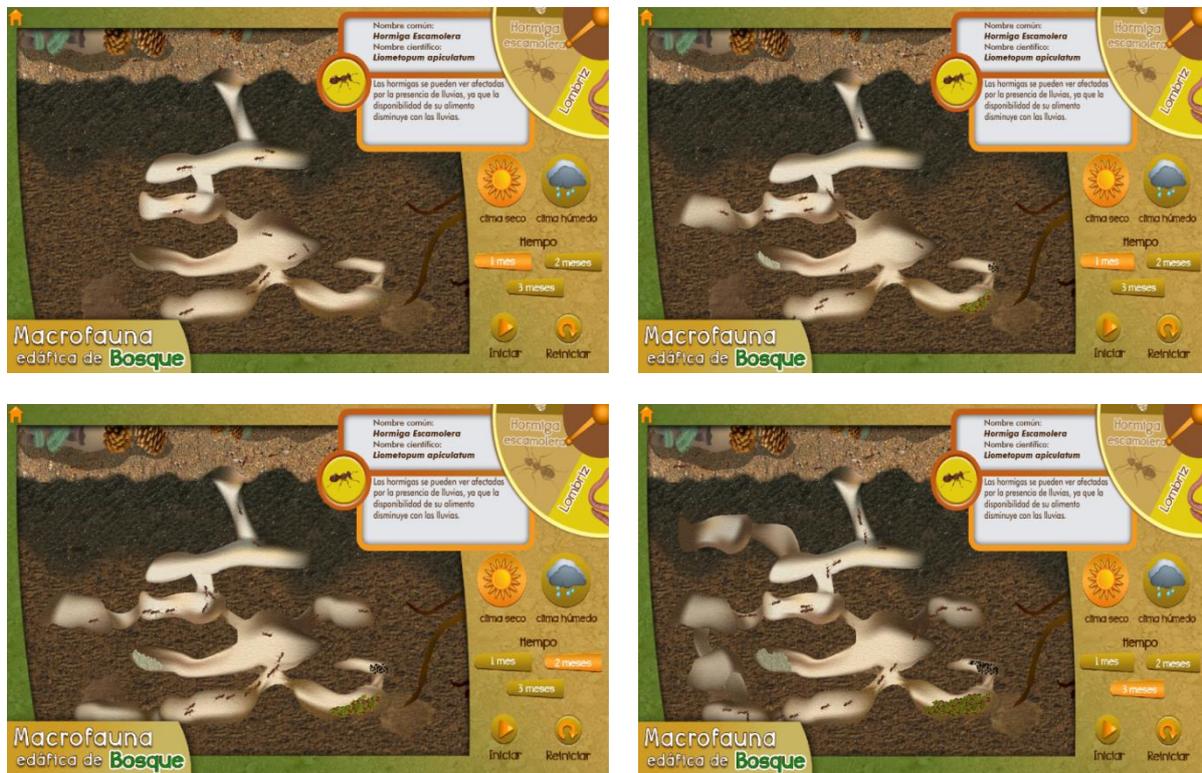


Figura 3.20 Simulaciones de hormiga escamolera.

El usuario tendrá la oportunidad de continuar haciendo cambios en el escenario cuantas veces quiera, o bien, regresar el microhábitat a su estado original tocando el botón reiniciar. Con la opción de inicio puede regresar a la pantalla de elección de ecosistema.

3.2.4 Productos generados

Como se describió en el capítulo anterior, durante esta fase se generan diferentes documentos que ayudan a la construcción del proyecto. En esta sección se describen los productos generados y se presenta un fragmento de ellos.

- Lista maestra: Es la lista que le asigna un nombre único a cada elemento multimedia, (ver figura 3.21) de esta manera es fácil identificar a un componente en específico. Cada área del equipo de trabajo puede tener su propia lista maestra.

Elemento	Tipo	Nomenclatura
	Botón	btn_climaHumedo_edo1

Figura 3.21 Fragmento de lista maestra.

- Interfaz gráfica: Se refiere a la vista gráfica final que tendrá la aplicación, incluyendo todas las ventanas principales, secundarias, ventanas emergentes y ventanas de ayuda. Además del estilo gráfico de los organismos.
- Mapa de navegación: Representa la funcionalidad de la aplicación a bajo nivel, es decir a través de un diagrama, se muestra la jerarquía, ordenamiento y navegación de pantallas del software (Ver figura 3.22).

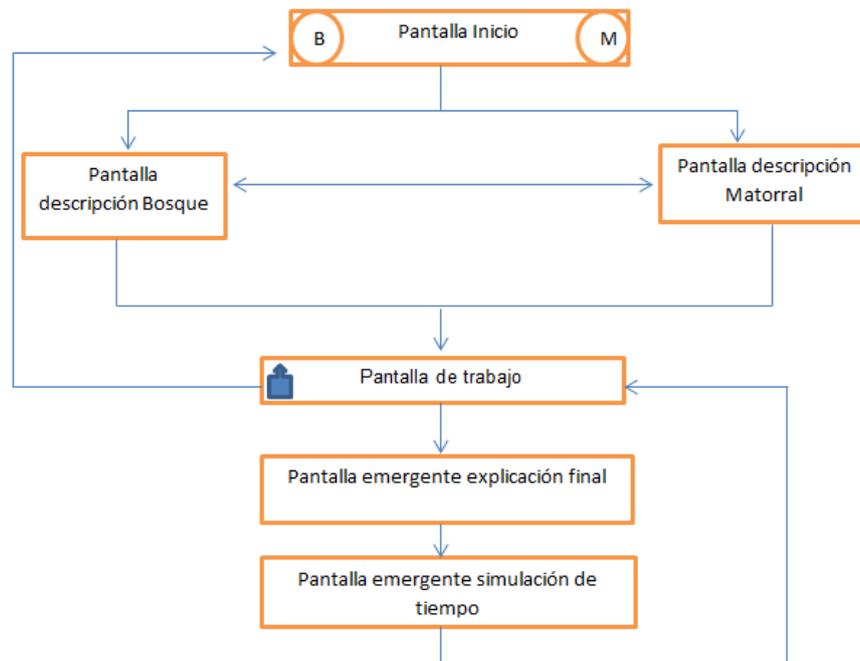


Figura 3.22 Mapa de navegación de Explora la función

- Guion funcional: El líder de desarrollo con base en la nomenclatura hecha en la lista maestra definirá la funcionalidad que hay en cada una de las pantallas de la aplicación, se especificará la función de los botones, imágenes o animaciones que contenga. Este documento es creado específicamente para el equipo de programación. La figura 3.23 presenta un fragmento del guion funcional de *Explora la función*.

Pantalla de elección de Ecosistema				
 <p>(No es imagen final)</p>				
	Descripción	Comportamiento	Tipo	Nomenclatura
8	Lupa Bosque Macrofauna Edo 1 en Inglés Lupa Bosque Macrofauna Edo 2 en inglés	El botón se encuentra en estado 1 cuando el usuario toque la lupa Bosque este se cambia a estado 2.	Botón	macro_lupaBosqueIng 1 macro_lupaBosqueIng 2
9	Botón de idioma español Edo 1 Botón de idioma español Edo 2	El botón de idioma se encuentra de lado Inferior Derecho, su función es cambiar de idioma según elija el usuario. Español e inglés.	Botón	p2_idioma1Esp p2_idioma2Esp

Figura 4.23 Ejemplo de guion funcional

- Guion multimedia: Se crea con base en la información de los documentos mencionados anteriormente. Representa una guía para todo el equipo de trabajo con el fin de que todos los involucrados sepan el comportamiento visual de la aplicación, a diferencia del guion conceptual aquí se deben de contar con las imágenes finales, describir a detalle el comportamiento de todos los elementos que tengan algún cambio o animación en cada pantalla y la duración de las animaciones. La figura 3.24 representa un fragmento del guion multimedia de *Explora la función*.

Pantalla de inicio	
Animación al toque	
<p>La animación de esta pantalla será que los organismo de los dos ecosistemas tengan movimiento: Hormiga: caminar dentro de la lupa. Araña: deberá moverse en el mismo espacio en el que se encuentra. Los botones de las banderas para la elección del idioma, deberán estar "parpadeando" para que el usuario sepa que puede cambiar el idioma, pero una vez que use alguno de estos botones se detiene la animación. Los botones cambian ligeramente de tamaño cuando aparecen por primera vez.</p>	

Figura 3.24 Ejemplo de guion multimedia.

3.3 Realización

Esta fase involucra la realización de contenidos, diseño y programación. La presente sección dejará de lado el desarrollo de contenidos y la forma de cómo se realizaron los gráficos en Photoshop y Illustrator, ya que no entran en el campo de estudio del presente trabajo. Esta sección se centrará exclusivamente en la realización e integración de elementos multimedia, animaciones y programación.

3.3.1 Arquitectura de Explora la función

Explora la función es una aplicación compleja que cuenta con muchos elementos multimedia, así pues, un punto muy importante en su realización fue la organización de todo este contenido. Para lograr esto se decidió estructurar el interactivo en tres pantallas principales: pantalla de inicio, pantalla de trabajo matorral y pantalla de trabajo bosque.

Como se ha mencionado en secciones anteriores Flash permite la creación de símbolos, estos son elementos de trabajo que permiten la manipulación del contenido multimedia. Durante el desarrollo del interactivo fue necesario estructurar dicho contenido mediante la creación de símbolos que ayudarán a modular el desarrollo y a tener una organización de los elementos dentro de la aplicación. En esta sección se describe cómo se encuentra la organización de los símbolos principales.

La figura 3.25 representa el símbolo de la pantalla de inicio, que encapsula el contenido de sus respectivos botones, títulos e imágenes, además contiene los símbolos de las pantallas secundarias: descripción de matorral y descripción de bosque. Estas pantallas secundarias cuentan en un nivel más interno con símbolos de botones e imágenes.

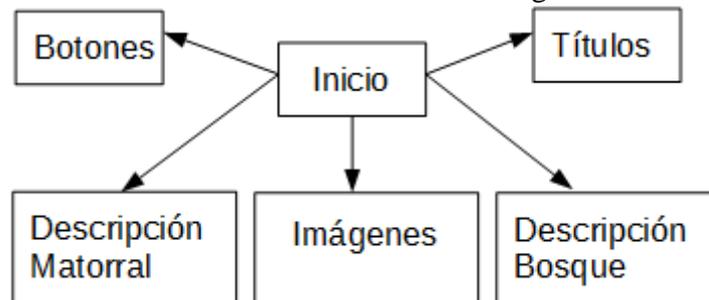


Figura 3.25 Símbolo Inicio

Para cada una de las pantallas de trabajo de matorral y bosque el contenido se separó en tres símbolos principales: *Contenedor Imágenes*, *Contenedor Botones* y *Contenedor Ruleta*.

La figura 3.26 muestra las imágenes principales de la pantalla de trabajo que están organizadas en el símbolo *Contenedor Imágenes*.

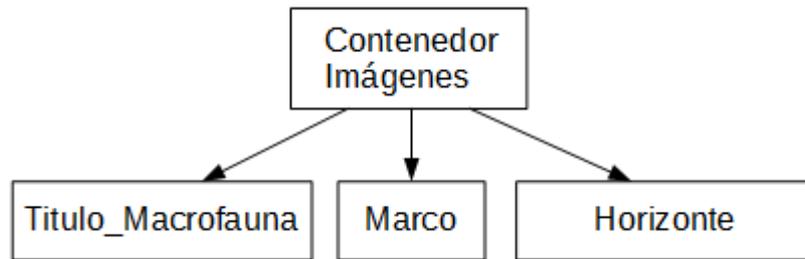


Figura 3.26 Símbolo Contenedor de imágenes

La ruleta es el principal elemento de interacción dentro de la pantalla de trabajo, es por ello que se decidió encapsular su funcionamiento en el *Contenedor Ruleta* y separar sus componentes (figura 3.27) para trabajarlos de manera independiente.

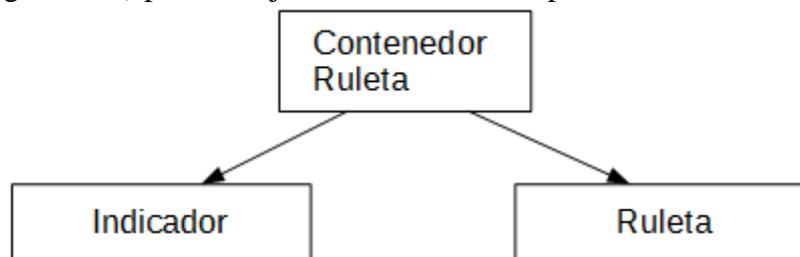


Figura 3.27 Símbolo Contenedor ruleta

Los botones de las pantallas de trabajo están en el *Contenedor Botones* (figura 3.28) cuentan con dos tipos de estado para indicar que han sido tocados por el usuario en la pantalla del dispositivo, para manejar estos estados hubo que trabajarlos por separado y hacerlos símbolos de tipo Botón.

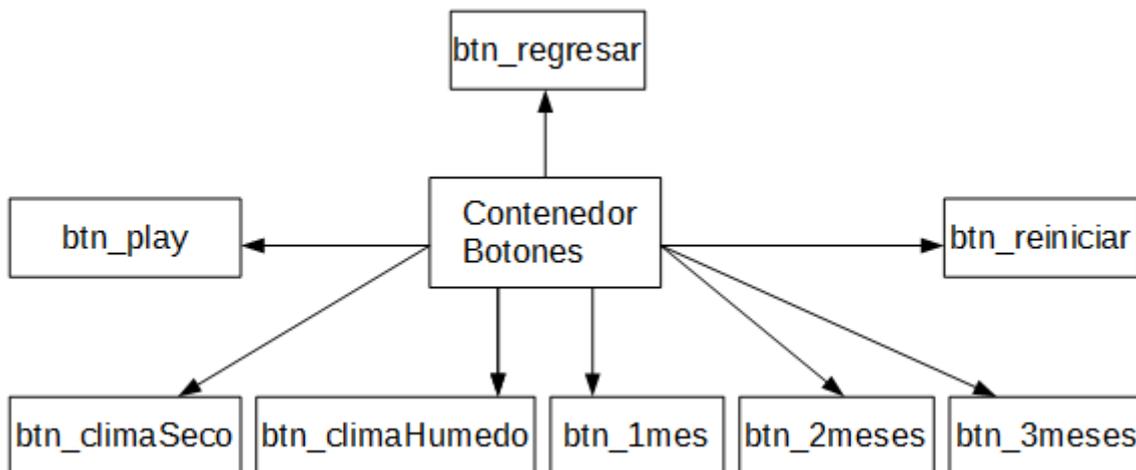


Figura 3.28 Símbolo Contenedor botones.

La idea general de esta organización es identificar los componentes principales de la aplicación para encapsularlos y separar botones de imágenes, convertirlas en símbolo y mostrarlas con base en el idioma con el cual se inicia la aplicación. Cabe mencionar que existen más símbolos dentro de los símbolos principales antes descritos, estos elementos son necesarios para realizar animaciones y otros aspectos visuales.

Teniendo propuesta la organización del contenido, el siguiente paso fue definir cómo y en qué momento tenían que aparecer en pantalla los símbolos. Esto se hizo a través del control de la línea de tiempo.

3.3.2 Organización de la línea de tiempo

La línea del tiempo organiza y controla el contenido multimedia a través del tiempo en fotogramas y capas. Un fotograma es una unidad de tiempo en Flash. Las capas se encuentran apiladas una sobre otra. Cada una de ellas puede contener uno o más símbolos. Cuenta con una cabeza lectora que se encarga de recorrer los fotogramas del símbolo e indicar que fotograma se está reproduciendo en un instante de tiempo. Ver figura 3.29

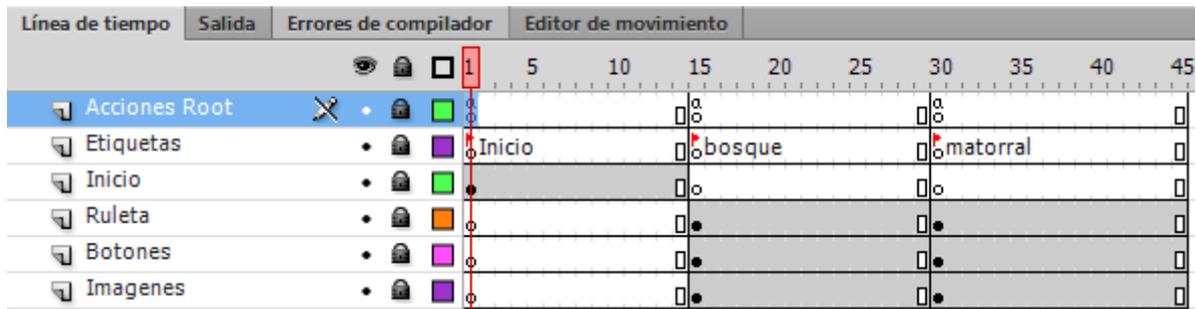


Figura 3.29 Línea de tiempo

La figura 3.29 representa la línea de tiempo principal de *Explora la Función*, en ella se encuentran las capas: Inicio, Ruleta, Botones e Imágenes. Estas capas contienen los símbolos descritos en la sección pasada. Las pantallas: Inició, Bosque y Matorral se separan mediante la capa etiquetas. La capa Acciones Root contiene el código ActionScript que se ejecuta cuando la cabeza lectora se encuentre en ese fotograma.

La pantalla inicial (ver figura 3.30) está encapsulada en el símbolo *Inicio* y contenida en la capa Inicio. Este símbolo contiene las pantallas secundarias de descripción bosque y matorral.



Figura 3.30 Desarrolló pantalla Inicio.

Flash puede asignar una línea de tiempo propia para un símbolo y separar en capas a los elementos que estén dentro de ese símbolo, gracias a esto se pueda trabajar con cada símbolo de manera individual en su respectiva línea de tiempo. Ingresando al contenido del símbolo Inicio se puede visualizar las pantallas secundarias de descripción de ecosistema.

Esto se muestra en las figuras 3.31 y 3.32.

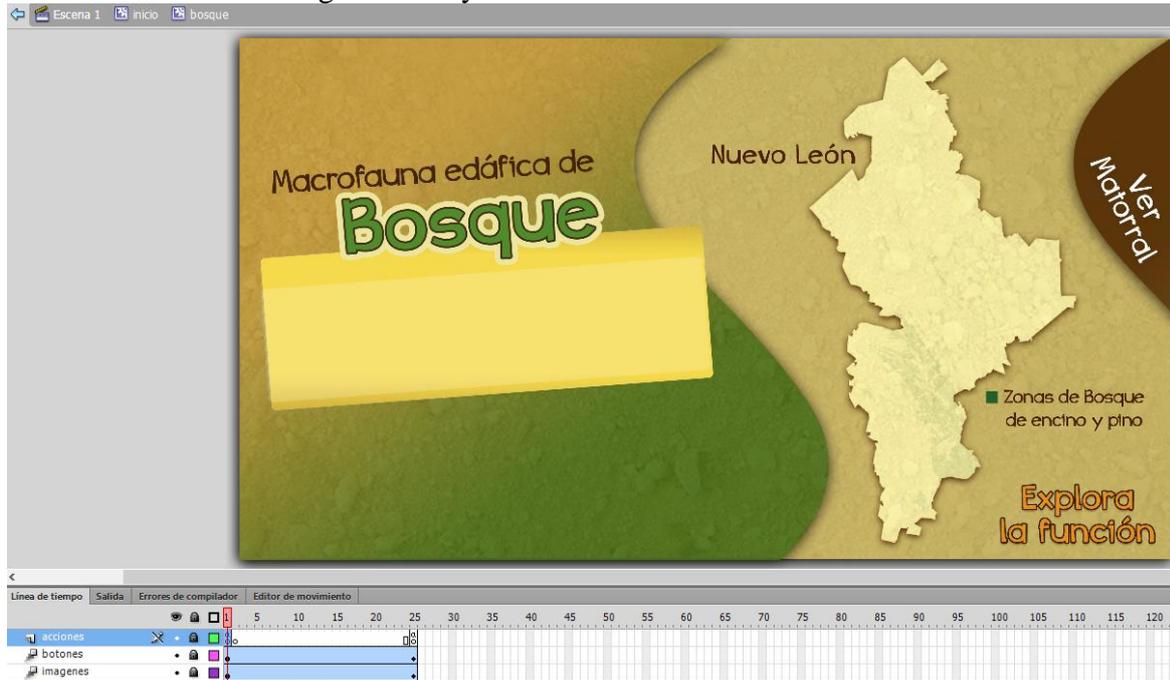


Figura 3.31 Pantalla Descripción Bosque.



Figura 3.32 Pantalla Descripción Matorral.

Las pantallas emergentes de explicación final y simulación de tiempo fueron diseñadas en símbolos distintos, cada uno de ellos cuenta con la estructura descrita anteriormente.

La pantalla de Bosque y Matorral se compone cada una de sus símbolos *Contenedor Ruleta*, *Contenedor Imágenes* y *Contenedor Botones*, esto se observa en las figuras 3.33 y 3.34.

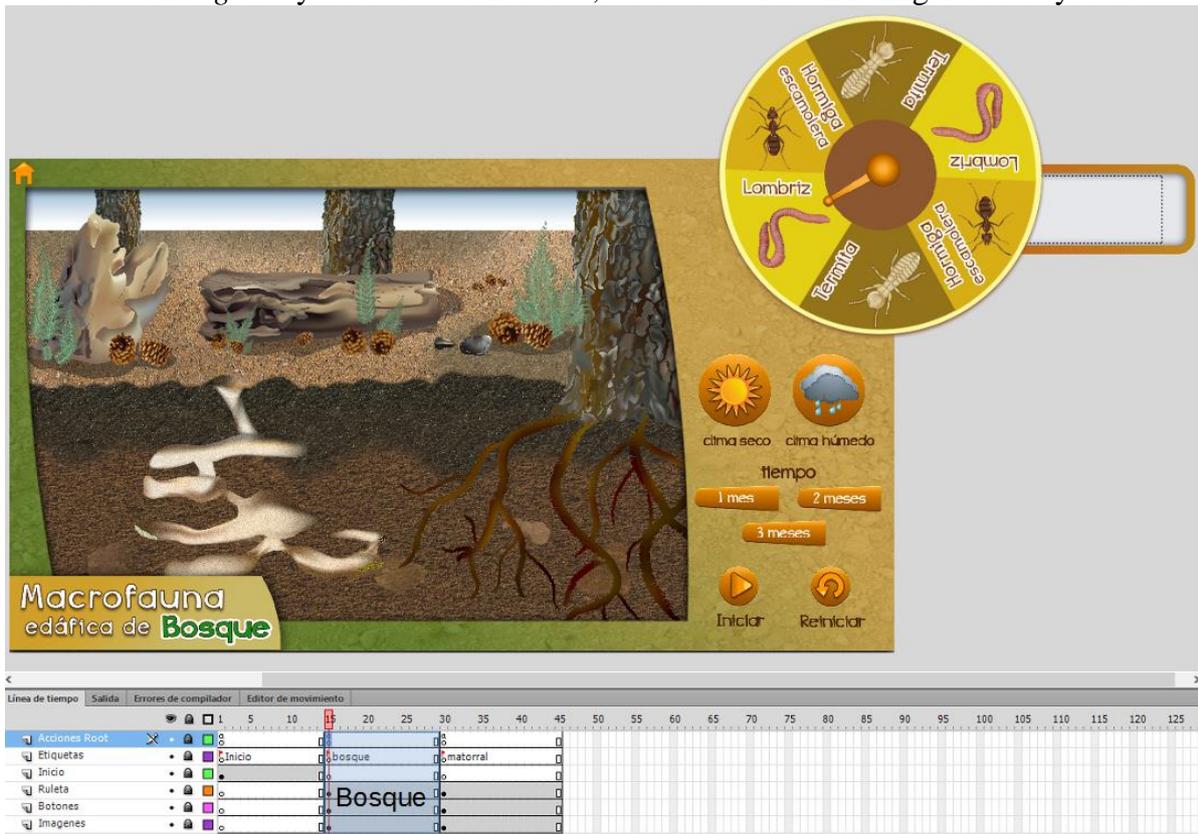


Figura 3.33 Desarrolló pantalla Bosque.

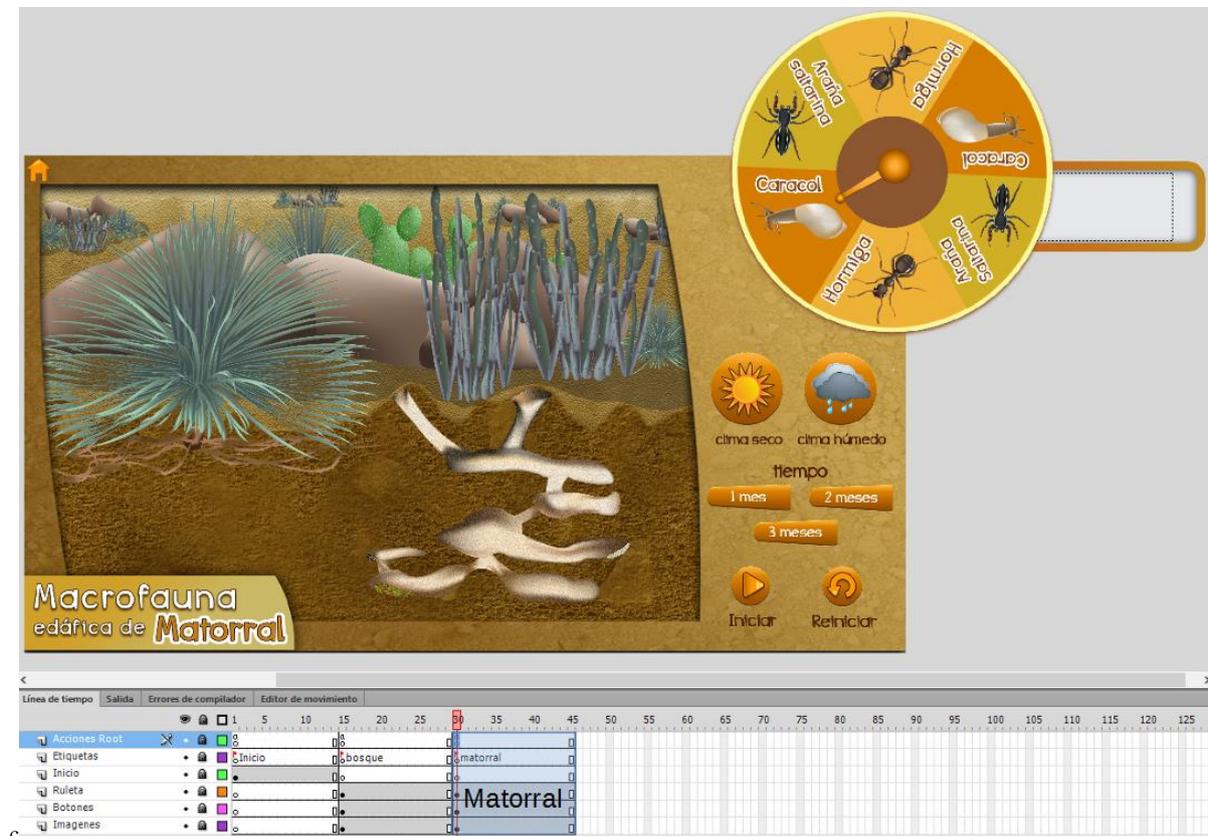


Figura 3.34 Desarrolló pantalla Matorral.

3.3.3 Animaciones

“La animación es el proceso de mover algo o, alternativamente, hacer que algo cambie a través del tiempo.” (Adobe Systems Incorporated. Programación con Adobe ActionScript, pág. 318). El tipo de animación usado en el interactivo se llama interpolación de movimiento y consiste en modificar las propiedades de visualización de los objetos (como la posición, el tamaño, la rotación o la propiedad alfa que representa la transparencia del objeto) fotograma a fotograma.

Una interpolación de movimiento comienza con un fotograma que contiene al símbolo en una posición inicial y un fotograma final que contiene un cambio en las propiedades del símbolo. La interpolación de movimiento calcula los valores de los fotogramas intermedios del símbolo en distintos estados y tiempos. Produce un efecto por el cual el primer estado evoluciona suavemente hacia el segundo, el segundo al tercero y así sucesivamente hasta terminar con los fotogramas que abarcan la interpolación.

Para *Explora la Función* fue necesario hacer muchas animaciones de este tipo en todas las pantallas, pero la principal muestra de la interpolación de movimiento está en la animación de los organismos interactivos, fueron un total 24 distintas animaciones de especies, 8 por cada escala de fauna. El reto principal fue emular los movimientos reales de las especies. Para ello, fue necesario buscar y ver videos de cada una las especies para poder replicar sus movimientos. De nueva cuenta la organización del contenido de diseño fue importante ya que había que determinar cómo separar en símbolos los elementos del cuerpo de la especie para poder trabajarlos de manera independiente de acuerdo al movimiento real, esto se puede observar en la figura 3.35 donde se tienen capas diferentes para los distintos elementos del cuerpo de la especie. Lo siguiente era animar a cada una tal que al combinar todas las partes se tuviera una sincronía y un movimiento natural en la animación.

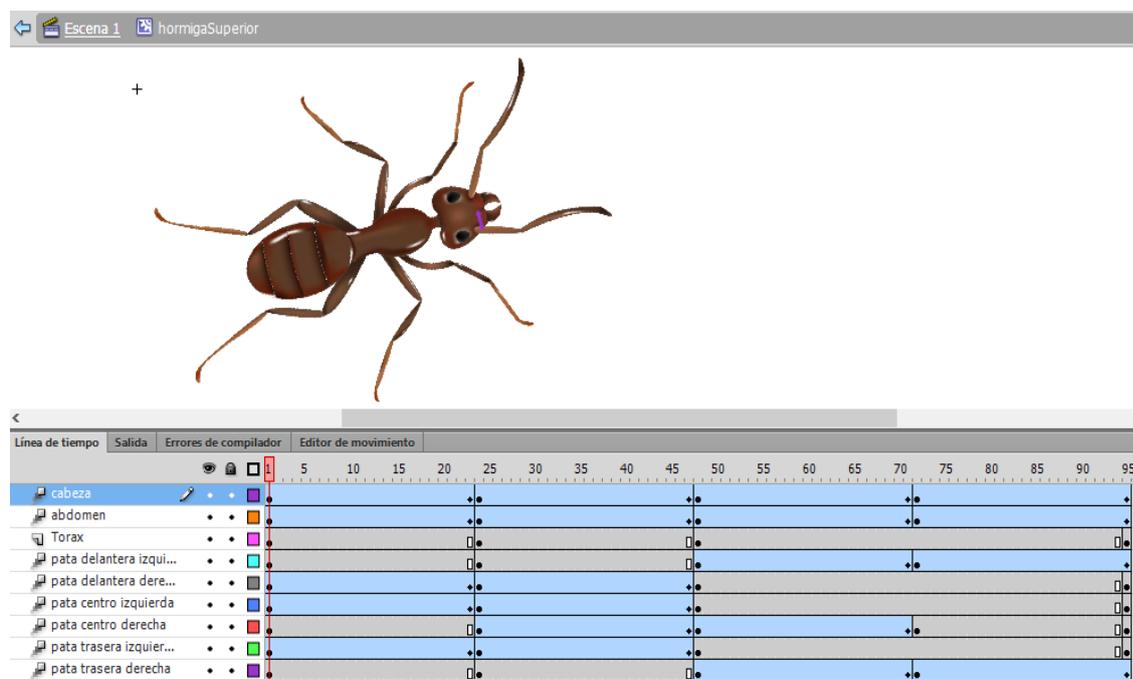


Figura 3.35 Interpolación superior de hormiga escamolera.

Al finalizar la animación todo el contenido se convierte a un símbolo de tipo clip de película y de esta manera podemos crear nuevos organismos interactivos que serán usado en otras pantallas del interactivo, por ejemplo cuando el usuario agrega especies al ecosistema.

Este proceso se repite para el resto de animaciones de organismos.

3.3.4 Código

El uso de símbolos no solo es importante para la organización del contenido sino también para trabajar los elementos a nivel de código ya que los objetos de tipo símbolo extienden de una clase llamada **MovieClip**⁴, esto permite que bajo la estructura de símbolos descrita en la sección 3.3.1 se pueda manipular los elementos de manera individual y modificar propiedades como **visible**, **X**, **Y**, **height**, **width**, etc., también nos permite usar funciones ya definidas por ActionScript o por el programador.

```
MovieClip(root).contenedor_imagenes.ilustracion_matorral.hormiguero2.gotoAndStop(1);  
MovieClip(root).contenedor_imagenes.ilustracion_matorral.hormiguero1.comida2.visible = false;
```

Figura 3.36 Uso de funciones y propiedades de la clase **MovieClip**.

La figura 3.36 muestra un fragmento de código donde se puede ejemplificar cómo mediante el diseño de la sección 3.3.1 se puede usar funciones como **gotoAndStop()** de la clase **MovieClip** que permite indicar a partir de que frame empezará la reproducción de la línea de tiempo. También muestra cómo modificar la propiedad **visible** de un símbolo específico.

La interacción en Flash es a través de eventos asignados a símbolos que se disparan al momento en que el usuario ejecute una acción, un ejemplo es la acción de tocar en pantalla un área que tenga alguna funcionalidad.

```
btn_1mes.addEventListener(MouseEvent.CLICK, function () {identificaTiempoPresiona(btn_1mes)});  
btn_1mes2.addEventListener(MouseEvent.CLICK, function () {identificaTiempoSuelta(btn_1mes2)});
```

Figura 3.37 Eventos para botones.

La figura 3.37 es un fragmento de código donde se muestra como mediante dos eventos distintos (MOUSE_DOWN y MOUSE_UP) se asigna una función al presionar y soltar el botón de clima. Estos eventos se utilizaron para detectar cuando el usuario tocaba en la pantalla del dispositivo algún elemento multimedia que tuviera cierta funcionalidad en el interactivo.

Además de las animaciones de interpolación, Flash permite realizar animaciones a través de código, para el caso de la animación de giro en la ruleta animada se utilizó una clase ya definida en la API de ActionScript llamada **Tween**. Mediante una instancia de esta clase podemos cambiar el tamaño, mover y desvanecer clips de películas especificando alguna de sus propiedades como la posición, tamaño o rotación.

⁴ MovieClip es una clase propia del lenguaje ActionScript la cual permite crear objetos de visualización en Flash.

El siguiente fragmento de código ejemplifica el funcionamiento de la ruleta animada.

```
var giraRuletaArriba:Tween;  
giraRuletaArriba = new Tween(ruleta,"rotation",Regular.easeOut,ruleta.rotation,ruleta.rotation + 60,0.7,true);  
giraRuletaArriba.start();
```

Figura 4.38 Animaciones mediante código.

En la figura 3.38 se declara una instancia de tipo **Tween** que se ejecutará cuando el usuario toque la ruleta y la desplace hacia arriba, los parámetros que recibe son: el nombre del símbolo, la propiedad rotación que se verá afectada, la función de movimiento que se utiliza de la clase **Regular**⁵, el valor inicial de la rotación, el valor final de la rotación, la duración que tendrá la animación y el valor booleano true que especifica el uso de segundos en vez de fotogramas.

Para iniciar la animación se hace uso del método **play()** ya definido por la clase **Tween**. Por cada que vez que se ejecute la animación se realiza una rotación de + 60 grados en el símbolo ruleta.

El uso de animaciones hechas en código basadas en la clase **Tween** fue implementado también en las ventanas de ayuda y la pantalla de simulación de tiempo con el reloj animado. La manera de hacerlas es análoga a la de la figura 3.38 pero variando la función de movimiento y la propiedad del símbolo que será afectada. El código de la animación se crea dentro de una función que después será ejecutada por un evento asociado a un símbolo específico.

Además de las animaciones mediante código, otra característica importante del interactivo al momento de la implementación fueron las simulaciones de cambios de ambiente.

El algoritmo que se usó para crear dicha funcionalidad se describe y ejemplifica con código ActionScript en los siguientes pasos:

1. Realizar el gesto de arrastre del organismo interactivo de la ficha informativa a la zona de su hábitat en la imagen del horizonte. La figura 3.39 muestra visualmente cómo hacer la acción de arrastre.



Figura 3.39 Arrastre del organismo al ambiente.

```
function arrastrar(me:MouseEvent, mc:MovieClip) {mc.startDrag();}  
function soltar(me:MouseEvent, mc:MovieClip) {mc.stopDrag();}
```

Figura 3.40 Código para el gesto de arrastre.

⁵ La clase Regular es una de varias clases de ActionScript que se ocuparon para utilizar funciones de aceleración para la implementación de movimiento en los símbolos.

La figura 3.40 muestra dos funciones que reciben como parámetro un símbolo de tipo movie clip que llamará a las funciones **startDrag()** y **stopDrag()** de ActionScript. Asignando las funciones arrastrar y soltar a un evento **MOUSE_DOWN** y **MOUSE_UP** como se mostró en la figura 3.37 tenemos la funcionalidad de arrastre para el símbolo de organismo interactivo.

2. Crear un organismo interactivo en su hábitat correspondiente cada vez que el usuario lo arrastre a su ecosistema.

Para este paso había que detectar cuando el organismo interactivo había sido arrastrado hasta su respectivo hábitat.

```
if (organismo.hitTestObject(habitat))
```

Figura 3.41 Detecta colisión con hábitat.

En la figura 3.41 la función **hitTestObject()** de ActionScript devuelve un valor booleano cuando dos símbolos colisionan, de esta manera cuando el usuario termina el arrastre y suelte el organismo se detecta si este está dentro del área de su hábitat. Si el valor de la condición es **true** se procede a crear el organismo interactivo.

La figura 3.42 ejemplifica la creación de una hormiga escamolera que está entrando al hormiguero después de que el usuario suelta al organismo en su hábitat.



Figura 3.42 Creación de hormiga escamolera.

Flash permite crear una clase de ActionScript y asociarla a un símbolo para generar instancias de esta mediante código.

```

public class RutaHormiga1 extends MovieClip
{
    public function RutaHormiga1(ejeX:Number,
                                ejeY:Number,
                                frame:Number,
                                idBicho:String)

    {
        this.x = ejeX;
        this.y = ejeY;
        this.gotoAndPlay(frame);
        this.idBicho = idBicho;
    }
}

```

Figura 3.43 Clase asociada a una ruta de hormiga.

La figura 3.43 muestra la clase RutaHormiga1 la cual cuenta con un constructor que recibe como parámetros la posición **X** y **Y** en donde se creará el organismo, el número de fotograma donde aparecerá en su línea de tiempo y el nombre del organismo.

Todos los organismos interactivos, a excepción de la lombriz de bosque, cuentan con una serie de rutas trazadas con capas guía en su línea de tiempo, la cual les indica hacia dónde dirigirse cuando son agregadas a su hábitat.

```

frame = Pelicula.dameAleatorio(1,340);
MovieClip(root).contenedor_imagenes.ilustracion_matorral.addChild(new RutaHormiga1(840,
                                                                                       891,
                                                                                       frame,
                                                                                       "Hormiga"));

```

Figura 3.44 Crear instancias de hormigas.

La figura 3.44 muestra la manera en cómo mediante el uso de clases asociadas a los símbolos de organismos interactivos se crean y agregan instancias de éste en la imagen del hábitat con el método **addChild()** de ActionScript.

3. Con base en la cantidad de organismos agregados, el clima y tiempo seleccionados por el usuario se debe realizar una simulación de cambio en el ambiente.

Los parámetros de clima y tiempo determinan el número de organismos que serán creados en el hábitat.

Por ejemplo si la especie en cuestión tiende a aumentar su población con un clima seco se toma la cantidad de elementos agregados por el usuario y se multiplica con múltiplos de dos como se muestra en la imagen 3.45, entre más especies agregue el usuario y mayor sea la cantidad de tiempo el número de organismos aumentará.

```

var numeroHormigas:Number = 0;
numeroHormigas = organismosAgregados * 2; //1 MES
numeroHormigas = (organismosAgregados * 2) * 2; //2 MESES
numeroHormigas = ((organismosAgregados * 2) * 2) * 2; //3 MESSSES

```

Figura 3.45 Cálculo de organismos para crear en el ecosistema.

En caso de que la población de organismos tienda a disminuir porque el parámetro de clima no es favorable para su reproducción la operación de cálculo de especies es dividir el número de especies agregadas por el usuario entre dos.

Finalmente dentro de un ciclo **for** se crean y agregan de manera aleatoria las distintas instancias del organismo con base en la variable **numeroHormigas**.

```
for (var i = 0; i < numeroHormigas; i++)
{
    ruta = Pelicula.dameAleatorio(1,4);
    if (ruta == 1)
    {
        frame = Pelicula.dameAleatorio(1,340);
        MovieClip(root).contenedor_imagenes.ilustracion_matorral.addChild(new RutaHormiga1(840,
                                                                                               891,
                                                                                               frame,
                                                                                               "Hormiga"));
    }
    if (ruta == 2)
    {
        frame = Pelicula.dameAleatorio(1,357);
        MovieClip(root).contenedor_imagenes.ilustracion_matorral.addChild(new RutaHormiga2(571,
                                                                                               450,
                                                                                               frame,
                                                                                               "Hormiga"));
    }
    if (ruta == 3)
    {
        frame = Pelicula.dameAleatorio(1,499);
        MovieClip(root).contenedor_imagenes.ilustracion_matorral.addChild(new RutaHormiga3(582,
                                                                                               453,
                                                                                               frame,
                                                                                               "Hormiga"));
    }
    if (ruta == 4)
    {
        frame = Pelicula.dameAleatorio(1,369);
        MovieClip(root).contenedor_imagenes.ilustracion_matorral.addChild(new RutaHormiga4(843,
                                                                                               904,
                                                                                               frame,
                                                                                               "Hormiga"));
    }
}
```

Figura 3.46 Cálculo de organismos para crear en el ecosistema.

La figura 3.46 muestra el fragmento de código que permite visualizar a las hormigas moviéndose en distintas direcciones dentro de su hábitat gracias a las instancias que se crean y que cuentan con su respectiva ruta a seguir. Los cambios visuales en el hábitat se realizaron con animaciones de interpolación o animaciones creadas con la clase **Tween** y se determina qué elementos se mostrarán con base en el número de organismos creados.

El código mostrado es específico de la hormiga escamolera, pero el algoritmo es aplicable para el resto de organismos.

La figura 3.47 es el resultado visual de implementar el algoritmo descrito anteriormente, se muestra un ejemplo de las rutas guía trazadas en el ambiente que permite a los organismos saber qué dirección tomar dentro del hormiguero.

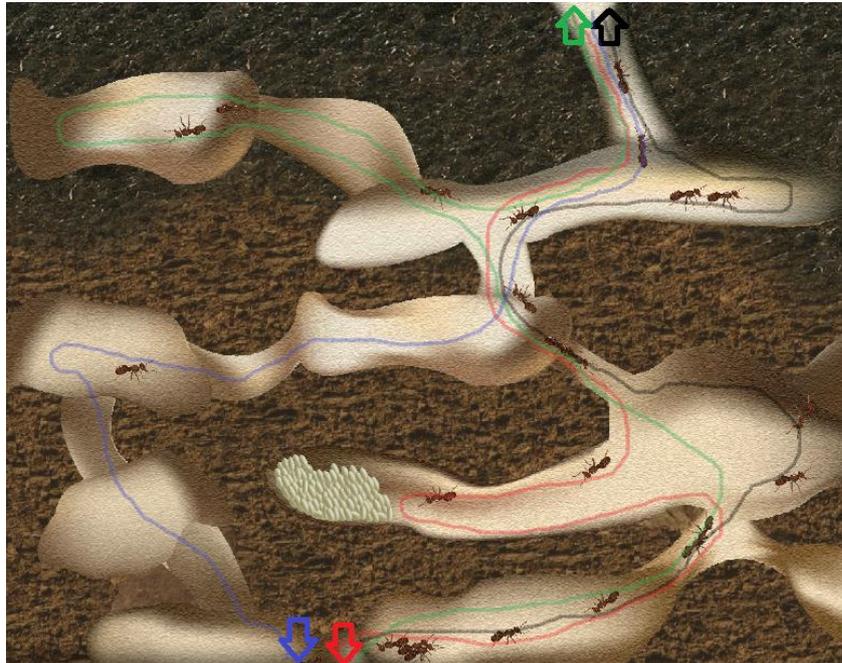


Figura 3.47 Representación visual al implementar el algoritmo.

3.3.5 Desempeño

Parte importante dentro del interactivo es el uso de animaciones, como se ha mencionado en secciones anteriores se realizaron animaciones mediante código e interpolación de movimiento. La necesidad de hacer animaciones complejas, como es el caso de los organismos interactivos implica un consumo considerable de recursos de hardware; en las primeras versiones, crear distintas instancias del organismo en una simulación provocaba que se disminuyera el desempeño del interactivo.

Dado que se requería una aplicación eficiente en cuanto a desempeño pero sin perder calidad en animación e imágenes, se trató de usar en lo posible la animación por código, ya que esto implica no usar la línea de tiempo y reducir el uso de memoria y procesador; sin embargo, la animación de organismos requiere una animación más detallada y la única manera de obtener dicho resultado es por interpolación de movimiento.

La manera de solucionar el problema de desempeño en las simulaciones fue asignar un límite a la cantidad de organismos creados, mediante una serie de pruebas con la aplicación se identificó una cota máxima de organismos a crear para cada especie. De esta forma no importa cuántos organismos agregue el usuario al ambiente, ni los parámetros de tiempo y clima que asigne a la simulación, mientras no se exceda la cota límite de organismos, se puede tener estabilidad en la aplicación en cuanto a rendimiento se refiere.

3.3.6 Contenidos

Los contenidos representan los textos que mostrará el interactivo. Un producto generado en esta fase es el guion de contenidos, el cual es realizado entre el experto y coordinador de contenidos, en él se incluye la información que mostrará cada pantalla en sus dos versiones de idioma: inglés y español.

Cabe mencionar que el interactivo cuenta con el manejo de textos de forma dinámica tanto para las versiones de inglés como en español, los cuales se cargan de un archivo XML. Esto permite que cuando se requiera cambiar la información solo es necesario modificar los contenidos en el archivo XML sin necesidad de volver a compilar o modificar el código del interactivo.

La imagen 3.48 muestra un fragmento del guion de contenidos de *Explora la función*. Se puede observar que todos los textos que están en pantalla cuentan con su respectiva información en inglés y español, algunos son textos estáticos y otros más son dinámicos como el caso del texto informativo (3), este contenido no es un texto final ya que puede ser modificado conforme el interactivo lo requiera, por lo cual se decidió que fuera leído de un archivo XML.

		Nombre: Descripción del ecosistema	Nomenclatura: p2_Bosque
	Nombre	Español	Inglés
1	Subtítulo de la aplicación	Macrofauna edáfica de Bosque	Forest edaphic macrofauna
2	Ver Ecosistema	Ver Matorral	See scrub
3	Texto informativo	Los bosques de pino y de encino abarcan el 11% de la superficie del estado y se ubican en lugares templados y húmedos. Están formados por árboles de hasta 22 m. de altura. Sus frutos son conos y bellotas, para pinos y encinos respectivamente.	The forests of pine and oak forests cover 11% of Nuevo Leon's area and are located in warm and humid places. They consist of trees up to 22 m. height. Its fruits are cones and acorns to pine and oak respectively
4	Nombre del Estado	Nuevo León	Nuevo León
5	Zonas	Zonas de Bosque de encino y pino	Areas of oak and pine forest

Figura 3.48 Fragmento de guion de contenidos.

3.4 Pruebas

Al finalizar la fase de realización por parte de los equipos de diseño, programación y medios, se obtuvo como resultado un prototipo de debía ser evaluado para garantizar que el trabajo hecho hasta el momento cumpliera con los objetivos definidos en la fase de planteamiento y análisis.

3.4.1 Evaluación con usuarios

Para identificar posibles problemas de usabilidad y de interacción, se realizó una serie de evaluaciones con usuarios en el laboratorio de observación del grupo ESIE ubicado dentro de las instalaciones del CCADET. En las evaluaciones participaron usuarios con un rango de edad de 7 a 16 años. En cada pantalla del interactivo se indicó a los usuarios que realizarán tareas relacionadas con la funcionalidad del interactivo. Adicionalmente, se realizaron dos cuestionarios para la evaluación, cabe señalar que los cuestionarios fueron diseñados para que a los usuarios les fuera sencillo expresar sus pensamientos y opiniones sobre el interactivo.

- Cuestionario de Usabilidad. Tiene como objetivo saber qué tan fácil era para el usuario realizar las actividades propuestas por el interactivo.
- Cuestionario de Satisfacción. Permite conocer la percepción que tenía el usuario del interactivo después de usarlo.

Los comentarios relacionados con las observaciones obtenidas de estos cuestionarios se presentan en la siguiente sección, como parte del análisis de los expertos.

Para apoyar la realización de este análisis y obtener resultados confiables sobre los problemas de usabilidad que presenta el interactivo, se grabó el comportamiento y comentarios de los usuarios ante las tareas que llevaron a cabo con el interactivo.

Con base en las observaciones realizadas en esta evaluación con usuarios, los expertos en evaluaciones del grupo de desarrollo elaboraron un reporte de las observaciones obtenidas pantalla por pantalla del interactivo.

A continuación se presentan los resultados:

- Pantalla inicio

Durante la evaluación se observó que los usuarios identificaron correctamente todos los elementos con los cuales pueden interactuar, por lo cual esta pantalla no reportó ningún cambio o problema.

- Pantalla descripción de ecosistema (matorral o bosque)

En general esta pantalla no reportó ningún problema de interacción con los usuarios, aunque sí se observó que no leían toda la información mostrada. Con base en esta observación, se solicitó al experto en contenidos que sintetizara la información presentada, para que fuera más concreta y a los usuarios les fuera más fácil leerla y comprenderla.

- Pantalla de trabajo

Los cambios en la pantalla de trabajo se definieron en gran medida por las observaciones en esta evaluación: se identificó que algunos usuarios tardaban en encontrar la forma de agregar las especies al escenario y otros más nunca encontraron la manera de hacerlo. Por lo cual, se decidió agregar una ayuda animada que ilustrará cómo agregar especies antes de que se comience con la actividad.

Algunos usuarios no identificaron que el botón *Reiniciar* sirve para volver el ecosistema a su estado inicial, ya que para realizar esta acción utilizaban el botón *Regresar* ubicado en la parte central superior de la pantalla y al tocarlo la aplicación los dirigía a la pantalla de inicio.

Adicionalmente a las evaluaciones con usuarios, a lo largo del proceso de desarrollo los expertos de contenidos, diseño e interacción fueron realizando diferentes evaluaciones de los prototipos construidos. En la siguiente sección se describe algunas de las observaciones realizadas por estos expertos.

3.4.2. Evaluación y análisis de los expertos

El proceso de desarrollo de software multimedia interactivo es un proceso iterativo, así que durante la realización de *Explora la función* se evaluaron distintos prototipos. En particular, en esta sección se describen algunas de las observaciones a la pantalla de trabajo, la figura 3.49 muestra una versión preliminar, en la cual se contemplaba que el usuario asignará los parámetros de acidez, humedad y temperatura mediante un gesto de arrastre en su respectivo indicador como la forma de establecer las características del ambiente.

Sin embargo los expertos detectaron que en las evaluaciones con usuarios no supieron definir para qué eran los controles de acidez, humedad y temperatura.



Figura 3.49 Versión inicial de la pantalla de trabajo.

La figura 3.50 muestra una comparativa visual de las distintas versiones de la pantalla de trabajo, las primeras contaban con los parámetros de acidez, temperatura y humedad, posteriormente se introdujeron los botones de clima seco y húmedo.

Dado los resultados de las pruebas se decidió modificar el estilo visual de los ecosistemas para distinguir uno de otro, además del diseño de la ficha informativa de la especie.



Figura 3.50 Comparación de versiones en la pantalla de trabajo.

Dadas las evaluaciones hechas se decidió realizar las siguientes modificaciones a la pantalla de trabajo:

- Los conceptos de acidez, humedad y temperatura son términos muy abstractos para los niños así que se decidió eliminarlos y buscar otra manera de representar los parámetros para la simulación. Con base en esto se eligió dejar solo los botones de *clima seco* y *clima húmedo*.
- Se sugirió agregar una ayuda más clara sobre el funcionamiento de la ruleta. Esto se resolvió con las ventanas de ayuda y una animación en la ruleta que captará la atención del usuario.
- Cambiar la ubicación del botón *Reiniciar*, para que los usuarios identifiquen la acción que se puede realizar al tocar este componente.
- Separar y sintetizar la información presentada en la ficha informativa para que a los usuarios les sea más fácil la lectura.

La versión final descrita en la sección 3.2.3 muestra la forma en que termino la organización de botones y la ficha informativa para solucionar estos dos últimos puntos.

Finalmente, se obtuvieron los siguientes resultados a partir de las repuestas dadas por los usuarios en los cuestionarios de satisfacción y usabilidad:

- El interactivo fue aceptable para los usuarios, el contenido de información es entendible, no es demasiada la cantidad de texto presentada en las pantallas del interactivo, las letras en pantalla fueron aceptadas, ya que les parecieron fáciles de leer y muy legibles.
- En la versión final del prototipo la organización de la pantalla fue adecuada para la mayoría de los usuarios, ya que fue fácil utilizar el interactivo, mientras que en las versiones iniciales una parte de los usuarios decía que no era adecuada la organización.
- Fue aceptable la forma en la cual se presentaron los botones en el interactivo, es decir su uso y relación con las acciones que representan.
- Las imágenes e ilustraciones presentadas en el interactivo fueron aceptadas por los usuarios, ya que les pareció adecuada la calidad y colores de las mismas.

Durante la revisión de los videos obtenidos de las evaluaciones con usuarios, se observó que la mayoría realizó durante el tiempo establecido las tareas indicadas en todas las pantallas, excepto en la pantalla de trabajo como ya fue descrito anteriormente.

3.5 Cierre

Al concluir las modificaciones de la aplicación surgidas en la fase anterior y generar una versión final tentativa del software, se realizaron las actividades para la entrega del interactivo.

El área de desarrollo creo manuales de uso y instalación para el cliente, el manual de uso describe la manera que el usuario debe realizar las actividades propuestas en el interactivo, el manual de instalación muestra de manera detallada los pasos para instalar el software en la tableta surface.

Como se ha mencionado en la sección de roles del capítulo anterior el experto en contenidos fue la persona encargada de realizar la investigación de contenidos que se muestra en el interactivo, dicha indagación quedó asentada en un reporte técnico el cual contiene información de los organismos, vegetación y suelo de Nuevo León. Este documento también fue entregado al cliente.

La investigación de los contenidos corrió a cargo del grupo ESIE, los avances se mostraban al cliente y si este aprobaba el contenido se procedía a adaptarlos a una versión para niños.

Ya que la investigación contiene muchos términos que pueden ser abstractos y de difícil comprensión para un niño que use el interactivo. Como parte del equipo de ESIE se contó con una persona especializada en la escritura de cuentos infantiles, la cual tuvo como tarea hacer la información más amena para los niños pero sin perder el contenido informativo.

La figura 3.51 muestra la comparativa de información de la ficha informativa de lombriz de tierra con base en el contenido del reporte técnico y la versión hecha por la especialista de escritura infantil.

	Propuesta basada en la investigación	Propuesta para niños
Características	Es una especie que puede alcanzar un tamaño de 9 cm, es muy pigmentada y nativa de Europa. Las lombrices son hermafroditas ya que poseen órganos reproductores masculinos y femeninos.	Esta lombriz viene de Europa puede medir hasta 9 cm, es de color rosa o ahí otras de color oscuro y pueden ser hembras o machos.
Clima	En invierno se entierran más para escapar de las heladas. Cuando el clima es muy caluroso, hacen lo mismo para evitar la deshidratación.	Cuando hace mucho frío como en invierno se meten lo más que pueden debajo de la tierra para no morir de frío. Y lo mismo hace cuando hace mucho calor para no morir de sed.
Explicación final	Esta lombriz es tolerante a suelos con altos niveles de metales pesados y tóxicos. Puede tolerar las condiciones ácidas de la basura de coníferas.	Esta lombriz puede vivir en todo tipo de suelo y aguanta cosas que los humanos no podríamos soportar y que nos harían daño.

Figura 3.51 Comparativa de textos de investigación y versión para niños.

La figura 3.52 muestra otro ejemplo de comparación entre el contenido de texto basado en la investigación del experto en contenidos y la versión hecha para niños.

Propuesta basada en la investigación	Propuesta para niños
Las especies de la macrofauna transforman las características físicas del suelo, por ejemplo, a través de la construcción de túneles. Algunas especies pueden triturar hojarasca y transforman las plantas muertas y otra materia orgánica en nutrientes para el suelo	¿Conoces túneles construidos por animales? ¡Las especies de la macrofauna son expertas! Logran maravillas al transformar las características físicas del suelo. Trituran hojarasca, convierten plantas muertas y otras materias orgánicas, en nutrientes para la tierra que habitan.

Figura 3.52 Comparativa de textos de investigación y versión para niños

En las figuras 3.51 y 3.52 podemos notar que el contenido propuesto para niños cuenta con términos más comunes y evita usar palabras que puedan no tener una semántica clara para el usuario, además de que con base en la información se hacen preguntas que despierten el interés por continuar la lectura.

Finalmente el grupo ESIE instaló el interactivo en las tabletas surface para que el cliente pudiera tener la última versión de *Explora la función* en un ambiente productivo.

3.6 Retroalimentación y Mantenimiento

Como se mencionó al inicio de este capítulo, *Explora la función* se compone de tres escalas: macrofauna, mesofauna y microfauna, cada una tiene la misma funcionalidad pero cuenta con organismos específicos, animaciones y simulaciones distintas para cada ecosistema.

En macrofauna el tamaño de los organismos en la realidad son visibles para la vista humana y la interacción de las especies con el ecosistema y los cambios que provocan en el ambiente fueron más fáciles de ejemplificar en el interactivo. Sin embargo para mesofauna y microfauna el tamaño de los organismos son microscópicos en la realidad, y mostrar los efectos que provocan en su hábitat se tornó más complejo dado el tamaño de las especies. Los cambios que provocan en el ambiente estos organismos son difíciles de percibir para el usuario. Parte del mantenimiento que se le dará a *Explora la función* es lograr que al usuario le sea más notorio dichos cambios.

Hasta el momento de escribir el presente trabajo el museo interactivo en el cual se exhibirá *Explora la función* aún no se encuentra abierto al público, esto ha permitido al grupo ESIE trabajar para mejorar la cuestión visual de cambios en el ambiente de mesofauna y microfauna.

3.7 Resumen

Toda la información mostrada en este capítulo representa el resultado general de aplicar varias iteraciones del proceso de desarrollo de software multimedia interactivo. El objetivo fue mostrar mediante un caso concreto la importancia de usar un proceso que vaya acorde a las características del software para desarrollar productos de calidad.

Al aplicar el proceso en *Explora la función* tenemos de manera general que:

La fase de planteamiento y análisis genera un documento en el cual se describe: Tema del interactivo, objetivos, perfil de usuario, contexto de uso, descripción del interactivo, especificaciones de hardware y software necesarios para su ejecución.

Con base en el guion conceptual en la fase de diseño se plantea una interacción simple la cual propone que el usuario realice ciertos gestos sobre la tableta donde se ejecutará el software y toque algunas zonas de la pantalla para realizar la actividad propuesta.

La fase de realización se centra en la elaboración de contenidos, diseño y programación.

Al terminar la realización del prototipo se continúa con la fase de pruebas, en donde se realizan evaluaciones con usuarios y con expertos al interactivo.

La fase de cierre contempla actividades finales tales como la instalación del prototipo en el hardware del cliente, el manual de usuario, manual de instalación y la investigación realizada de los contenidos.

Finalmente la etapa de retroalimentación y mantenimiento toma en cuenta que hay ciertas modificaciones por hacer al software, dichas modificaciones son parte del mantenimiento que el grupo ESIE contempla realizar a futuro.

4 Conclusiones

4.1 Resumen

En este trabajo se describe al software como algo intangible, de manera más específica como un conjunto de programas formados por una serie de instrucciones que tiene que ejecutar una computadora, dada su gran importancia en la actualidad se plantea que este debe ser de calidad, para lograr este objetivo existe la ingeniería de software la cual define un marco de trabajo basado en procesos, herramientas y métodos. El proceso indica al equipo de trabajo qué es lo que hay que hacer, quién, cómo y cuándo hay que hacerlo. Los métodos son los aspectos técnicos que permiten el desarrollo de software y las herramientas dan el soporte a los métodos y el proceso.

El núcleo de la ingeniería de software se encuentra en el proceso, teniendo en cuenta la gran variedad de software que existe en la actualidad es imposible definir un proceso general que permita desarrollar cualquier tipo de software. Con base en esto las personas que se encargan de desarrollar software deben de tener sus procesos definidos y adecuarlos a las necesidades del software tomando como base un modelo. En este trabajo se plantea que un modelo es una representación abstracta de un proceso.

En el capítulo 1 explicamos los modelos de software más usados en la actualidad y en el cual tomando las actividades de un modelo base, se extiende o agregan ciertas características para crear modelos más específicos. Se mostró que todo proceso de software está basado en ese modelo básico el cual tiene cuatro actividades o fases genéricas: Especificación de requerimientos, diseño e implementación del software, validación del software y evolución del software. Estas actividades son complejas y están formadas por un conjunto de tareas, las cuales deben realizarse por la o las personas que tengan un rol con esas responsabilidades. Vimos que los procesos iterativos son los procesos más usados en la actualidad ya que están diseñados para solventar los eventuales cambios que existen en el desarrollo del software y por este motivo, para el desarrollo de este proyecto se toma como base esta forma de trabajo.

Con esto en mente, describimos el proceso de desarrollo de software multimedia interactivo, un proceso iterativo que es usado por el grupo ESIE para el desarrollo de software, el cual está formado por seis fases: Planteamiento y análisis, diseño, realización, pruebas y evaluación, cierre y como última fase retroalimentación y mantenimiento. Para poder seguir este proceso fue necesario entender cada una de sus fases, sus roles, tareas y productos generados.

Finalmente haciendo uso del proceso de desarrollo de software multimedia interactivo se describe la creación de *Explora la función* un software multimedia hecho por el grupo ESIE para el museo “Papalote Verde” ubicado en Monterrey Nuevo León. El software tiene como objetivo mostrar al suelo como un microhábitat en donde habitan diferentes especies y además observar los cambios que ocurren en este microhábitat con base en las características del suelo y fauna que lo habita. El software cuenta con animaciones, video, texto e imágenes que en conjunto ofrecen una interacción al usuario para que mediante gestos o toques en la pantalla de un dispositivo móvil pueda visualizar los cambios que provocan ciertos organismos en el ambiente con base en los parámetros de clima y tiempo que el usuario elija.

En el trabajo se describe a detalle cómo se aplicó el proceso fase por fase y como con base en los productos generados durante el proceso de desarrollo, el interactivo fue evolucionando de un prototipo burdo hasta llegar a ser un interactivo que será exhibido en el museo “Papalote Verde”. Esto nos permitió obtener un prototipo probado por usuarios con los criterios de calidad definidos.

4.2 Observaciones finales

Es innegable la importancia que tiene el software en la actualidad, en mi experiencia como programador fuera del ámbito académico y laborando en un par de empresas privadas he tenido la oportunidad de trabajar en un sistema web y en aplicaciones de escritorio que se ejecutan en equipos remotos. Comparando el proceso que se discutió en este trabajo con la forma de desarrollar software en dichas empresas, puedo decir que he estado lejos de seguir un proceso en el mundo laboral. Si bien, he usado metodologías y herramientas, no un proceso completo. Es por esto que la experiencia que viví con el grupo ESIE fue muy productiva para mí ya que es la primera vez que trabajo en un software multimedia interactivo y veo la importancia que tienen los procesos de software en proyectos reales y complejos. Puedo afirmar que sin el proceso no hubiera sido posible saber si lo que se realizaba era correcto y como consecuencia habría resultado muy difícil realizar el interactivo, esto a la larga provoca pérdidas de tiempo y dinero, dos conceptos que en la actualidad son de suma importancia para una persona y no se diga para las compañías.

Gracias a este trabajo pude enriquecer mis conocimientos en ingeniería de software, aprender un nuevo lenguaje de programación y comprender que para hacer software de calidad no basta solo con implementar buen código, es necesario contar con un proceso que respalde el desarrollo.

La opinión personal que tiene este servidor sobre el proceso que se usó en el presente trabajo, es que es un proceso correcto y eficiente, sin embargo, con base en la experiencia vivida en *Explora la función*, considero que la fase de cierre podría ser omitida ya que al ser un proceso iterativo en donde las distintas áreas trabajan concurrentemente en cada fase, es posible crear los manuales de usuario y de instalación junto con el reporte técnico y realizar la instalación del sistema en la fase de retroalimentación y mantenimiento. Esto permitiría reducir las actividades del proceso y hacer iteraciones más cortas.

Además, si consideramos que al dejar abierta la posibilidad de cambios en la fase de retroalimentación y mantenimiento, el número de iteraciones de esta etapa parece indefinido. Creo que con base en el mantenimiento que se tenga que hacer al software, el área de desarrollo debe acotar el número de iteraciones para que a la brevedad considere terminado el interactivo.

4.3 Trabajo a futuro

El museo Papalote Verde está próximo a inaugurar la zona donde se exhibirá Explora la función por tal motivo las modificaciones que se tienen pendientes en la fase de retroalimentación y mantenimiento se deben resolver en un corto plazo. Además se contempla que una vez que el interactivo esté en uso dentro del museo, se monitoreara el funcionamiento y, en caso de surgir problemas de desempeño, cambios en la interactividad o que surja algún error que no haya sido detectado en las pruebas, se corrijan a la brevedad.

Anexo A

Para cada especie fue necesario realizar simulaciones que mostraran los cambios de ambiente provocados por dichos organismos con base en los parámetros de clima y tiempo asignados por el usuario. A continuación se describe el diseño de las simulaciones y sus características principales.

Cambios de ambiente en *Matorral*.

- **Hormiga negra:** Cuenta con cuatro cambios de ambiente, la principal característica es el incremento de la colonia y el hormiguero, las hormigas aumentan con un clima seco y disminuyen con el húmedo, siguen rutas guía hechas en la línea de tiempo, lo cual les permite desplazarse para salir y entrar en el hormiguero.



- **Caracol terrestre:** Cuenta con tres cambios de ambiente, los caracoles tienen rutas guía para desplazarse sobre las hojas de la planta, fue necesario realizar el efecto de que ingresaran por las hojas centrales y salieran por las hojas traseras para poder tener una animación más apegada a la realidad. Con base en la cantidad de caracoles generados estos consumen la planta. La población crece con un clima seco y disminuye con el húmedo.

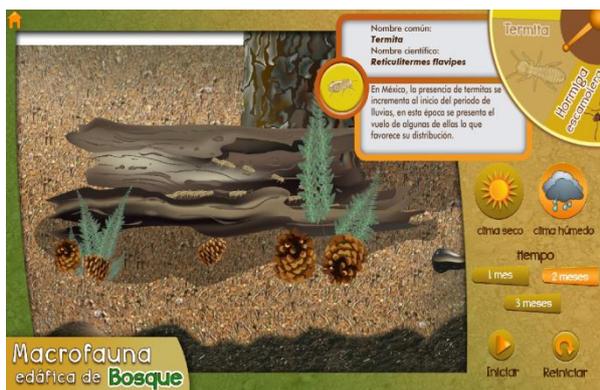
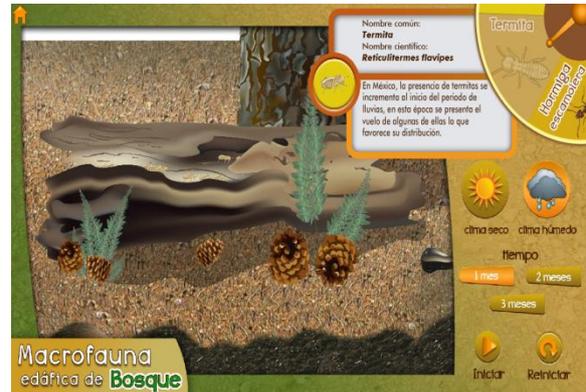
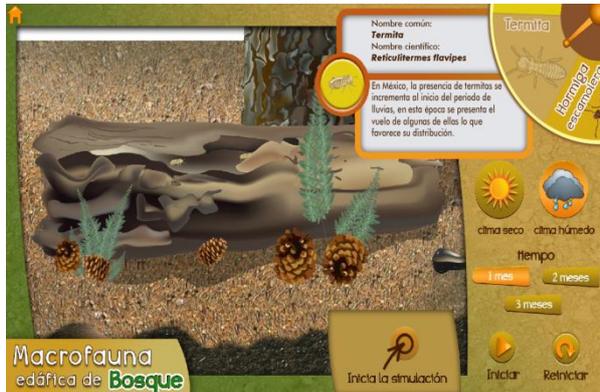


- Araña saltarina: Tienen rutas trazadas alrededor de la planta, fue necesario realizar efectos de salto mientras siguen la ruta ya que esa es una característica especial de este organismo. Si la población crece debilitan la planta y la van secando poco a poco. Su población aumenta en clima seco y disminuye con el húmedo.



Cambios de ambiente en Bosque.

- **Termita:** Para este organismo se realizaron cuatro cambios de ambiente el principal aspecto visual fue el tronco de madera y el efecto de que las termitas lo consumen conforme pasa el tiempo. Las termitas siguen distintas rutas en el tronco para cada cambio de ambiente fue necesario realizar el efecto de que el organismo camina en el interior del tronco para tener un simulación más realista. Su población aumenta con clima húmedo y disminuye con el seco.



- **Lombriz:** Los cambios de ambiente para esta especie fue el crecimiento de las raíces del árbol y el hundimiento de las rocas en el suelo. Este cambio resultó ser un poco más sencillo que los demás dado que las lombrices están animadas pero no cuentan con una ruta a seguir, permanecen en el mismo punto donde se agregan.



Bibliografía:

Adobe Systems Incorporated. Programación con ADOBE ACTIONSCRIPT 3.0, (2008)

Visitado en: 2 de diciembre de 2016,

http://help.adobe.com/es_ES/ActionScript/3.0/ProgrammingAS3/flash_as3_programming.pdf

Adobe. Adobe Flash Professional Ayuda y Tutoriales. (2014)

Visitado en: 2 de diciembre de 2016,

https://helpx.adobe.com/es/pdf/flash_reference.pdf

Adobe. Help Adobe. Uso de Adobe Flash Professional CS5 y CS5.5. (2011)

Visitado en: 2 de diciembre de 2016,

http://help.adobe.com/es_ES/flash/cs/using/flash_cs5_help.pdf

Belloch C. Aplicaciones Multimedia. Unidad de Tecnología Educativa (UTE). Universidad de Valencia. Visitado en: 2 de diciembre de 2016

<http://www.uv.es/bellohc/logopedia/NRTLogo4.pdf>

De la Cruz, G., Eslava, A., Castañeda, R. (2014) Proceso de desarrollo de software para aplicaciones multimedia interactivas (Notas de clase) CCADET, UNAM.

De la Cruz Martínez, G, Eslava Cervantes, A. y Castañeda Martínez, R. (2015) Diseño de la Experiencia del Usuario para Espacios Interactivos de Aprendizaje no Formal, Research in Computer Science, 89, p. 53–62.

De la Cruz, G., Eslava A y Castañeda R. (2014) Evaluaciones de usabilidad del interactivo “Suelos y Biodiversidad”. CCADET, UNAM.

Ibargüengoitia, G., y Oktaba, H. (2010). Ingeniería de Software Pragmática.

Ibargüengoitia, G., Levy, S. y Ramírez, A. (2008) Proceso de producción y administración de proyectos multimedia.

López A. (2005) Manual de Edafología. Departamento de Cristalografía, Mineralogía y Química Agrícola de la Universidad de Sevilla. .

Macromedia, Inc. Primeros pasos con Flash (2005).

Visitado en: 2 de diciembre de 2016.

https://ocw.innova.uned.es/mm2/tm/contenidos/pdf/tema6/Flash8_PPasos.pdf

Oktaba, H., Esquivel, C. A., Ramos, A. S., Martínez, A. M., Osorio, G. Q., López, M. R., ... & Lemus, M. Á. F. (2005). Modelo de procesos para la industria de software Moprosoft. Secretaría de Economía, México.

Visitado en: 2 de diciembre de 2016.

<http://kualikaans.fcencias.unam.mx/images/kuali/documentos/MoProSoft.pdf>

Pressman R. (2005) Ingeniería del Software. Un enfoque práctico. McGraw Hill 6ta. Ed.

Sommerville I. (2011) Ingeniería de Software. Pearson 9na. Ed.