

Universidad de Cienfuegos “Carlos Rafael Rodríguez”

Facultad de Informática

Carrera de Ingeniería Informática

Trabajo de diploma para optar por el título de Ingeniería en Informática

**Sistema para el control de Especies y Familias Botánicas del
Jardín Botánico Oasis**

Autor:

Enmanuel Velazco Martínez

Tutor:

Ms. C. Laura Toledo. Universidad de Cienfuegos.

Cienfuegos, Cuba

Curso 2005 - 2006

A mi familia, mi novia y mis amigos
Muchas gracias por su apoyo.
Que Dios los bendiga como me bendijo a mí por tenerlos a ustedes.

Resumen

El Sistema de control de especies y familias botánicas del Jardín Oasis es un sistema informático que permite el control de las familias y las especies botánicas existentes en el jardín Oasis. Este sistema almacena los datos de las especies y las familias, así como la ubicación de las especies botánicas en un mapa del jardín para tener un control más detallado de las mismas. La idea de este sistema surge dado el ineficiente control sobre las especies que existen en el jardín Oasis. Se procedió entonces a la búsqueda de información de herramientas existentes y al no encontrar otras que satisficieran con los objetivos concretos del jardín se acordó entonces la creación de este sistema informático. El sistema ofrece la posibilidad de manipular la información de las especies y las familias, además permite la importación y exportación de la base de datos a archivos XML y Excel, también permite la ubicación de especies en un mapa, y la generación de índices que son de complejo cálculo y reportes.

Índice

| | |
|--|-----------|
| <i>Introducción.....</i> | <i>1</i> |
| <i>Capítulo I: Marco Teórico.....</i> | <i>6</i> |
| Introducción | 6 |
| Descripción del dominio del problema..... | 6 |
| Botánica..... | 6 |
| Jardín Botánico..... | 7 |
| Descripción del objeto de estudio | 8 |
| Descripción del entorno..... | 8 |
| Herramientas Existentes..... | 9 |
| Descripción del sistema actual | 10 |
| Descripción del objeto de automatización..... | 10 |
| Fundamentación de las metodologías a utilizar | 10 |
| UML | 10 |
| Proceso Unificado de desarrollo (RUP) | 12 |
| Fundamentación de las herramientas y lenguajes utilizados | 13 |
| Arquitectura de tres capas..... | 13 |
| C++..... | 14 |
| Borland C++ Builder 6 | 16 |
| Java..... | 19 |
| SQL | 22 |
| Fundamentación del gestor de bases de datos utilizado | 23 |
| Microsoft Office Access 2003..... | 23 |
| MySQL..... | 24 |
| Conclusiones del capítulo | 24 |
| <i>Capítulo 2 Modelo del Dominio y Requisitos.....</i> | <i>26</i> |
| Introducción | 26 |
| Definición de las entidades y los conceptos principales | 26 |
| Reglas del negocio a considerar | 27 |

| | |
|---|------------------|
| Representación del modelo del dominio..... | 27 |
| Descripción del sistema propuesto..... | 27 |
| Concepción general del sistema | 27 |
| Requerimientos funcionales | 29 |
| Requerimientos no funcionales | 30 |
| Modelo de casos de uso del sistema | 31 |
| Actores del sistema..... | 32 |
| Diagramas de casos de uso del sistema | 32 |
| Descripción de los casos de uso del sistema..... | 33 |
| Conclusiones | 38 |
| <i>Capítulo 3 – Construcción de la solución propuesta.....</i> | <i>39</i> |
| Introducción | 39 |
| Diagrama de clases del diseño..... | 39 |
| Diseño de la base de datos | 41 |
| Modelo lógico de datos | 41 |
| Modelo físico de datos..... | 41 |
| Diagrama de implementación | 42 |
| Principios de diseño | 43 |
| Estándares en la interfaz de la aplicación | 43 |
| Tratamiento de errores..... | 44 |
| Concepción General de la ayuda | 44 |
| Conclusiones | 44 |
| <i>Capítulo 4 Estudio de Factibilidad.</i> | <i>45</i> |
| Introducción | 45 |
| Planificación por Puntos de Función..... | 45 |
| Costos | 48 |
| Beneficios Tangibles e Intangibles..... | 52 |
| Análisis de Costos y Beneficios | 52 |
| Conclusiones del capítulo | 53 |
| <i>Conclusiones</i> | <i>54</i> |

| | |
|---|----|
| <i>Recomendaciones</i> | 55 |
| <i>Referencias Bibliográficas</i> | 56 |
| <i>Bibliografía</i> | 58 |
| <i>Anexos</i> | 59 |
| Prototipos..... | 59 |

Índice de tablas

| | |
|--|----|
| Tabla 1 Descripción del caso de uso Gestionar Familias Botánicas | 34 |
| Tabla 2 Descripción del caso de uso Gestionar Especies Botánicas | 35 |
| Tabla 3 Descripción del caso de uso Mostrar Reportes | 35 |
| Tabla 4 Descripción del caso de uso Imprimir Reportes | 36 |
| Tabla 5 Descripción del caso de uso Importar Datos | 37 |
| Tabla 6 Descripción del caso de uso Exportar Datos | 37 |
| Tabla 7 Descripción del caso de uso Mapear Especies | 38 |
| Tabla 8: Entradas Externas | 46 |
| Tabla 9: Salidas Externas..... | 46 |
| Tabla 10: Peticiones | 47 |
| Tabla 11: Ficheros Internos | 47 |
| Tabla 12: Puntos de Función..... | 48 |
| Tabla 13: Miles de Instrucciones Fuentes | 48 |
| Tabla 14: Multiplicadores de Esfuerzos y Factores de Escala..... | 50 |
| Tabla 15: Resultado de los Costos..... | 52 |

Índice de Figuras

| | |
|--|----|
| Figura 1 Modelo de Dominio..... | 27 |
| Figura 2 Diagrama de casos de uso del sistema..... | 33 |
| Figura 3 Diagrama de clases de diseño | 40 |
| Figura 4 Modelo lógico de datos..... | 41 |
| Figura 5 Diagrama del Modelo físico de datos | 42 |
| Figura 6 Diagrama de Implementación..... | 43 |
| Figura 7 Prototipo Familias..... | 59 |
| Figura 8 Prototipo Insertar Familia | 60 |
| Figura 9 Prototipo Modificar Familia | 60 |
| Figura 10 Prototipo Ver Familia | 61 |
| Figura 11 Prototipo Especies..... | 61 |
| Figura 12 Prototipo Insertar Especie | 62 |
| Figura 13 Prototipo Modificar Especie | 62 |
| Figura 14 Prototipo Ver Especie..... | 63 |
| Figura 15 Prototipo Reportes..... | 63 |
| Figura 16 Prototipo Mostrar Mapa | 64 |

Introducción

La actual revolución científico-técnica y el proceso de globalización que están teniendo lugar en el mundo contemporáneo, conjuntamente con la alta competitividad e incertidumbre de las economías, exige cada vez más el desarrollo de la informática y el auge del conocimiento, como factores decisivos en la búsqueda de soluciones alternativas, tanto para la economía global como local, así como para la sociedad en general.

El avance en las tecnologías de la información es sin duda una de las grandes herramientas que está determinando el avance técnico de la humanidad, ya sea a niveles de investigación, ciencia, educación, comunicación, gobierno, o de comercio. Es por ello que se debe estar atento a estos avances ya que quizás alguno sea el que determine la manera en que se trabaja, se aprende o se socializa en un futuro no muy lejano.

Las TIC (Tecnología de la Información y las Comunicaciones) han venido a marcar una nueva era en la gestión de la información. Las tecnologías de la información y la comunicación son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario. Se denominan Tecnologías de la Información y las Comunicación al conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las TIC incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual[1].

Conocimiento, innovación y capacidad de aprendizaje son tres aspectos complementarios del desenvolvimiento actual de las sociedades avanzadas.

La instrumentación tecnológica es una prioridad en la comunicación de hoy en día. Las tecnologías de la comunicación son una importante diferencia entre una civilización desarrollada y otra en vías de desarrollo.

En este proceso de cambio estructural de las sociedades modernas se destaca, al menos, una dimensión: la generación de nuevos avances científicos y, especialmente, la difusión de nuevas tecnologías, singularmente, las TIC; esto hace que el proceso de informatización de las empresas se esté convirtiendo en un hecho cada vez más necesario. Anteriormente, los equipos y sistemas se han utilizado, por lo general, como elementos informatizados aislados, con escasa o nula comunicación entre sí en la mayoría de los casos; sin embargo, en la actualidad se ha iniciado un proceso encaminado a la fusión e integración a través de redes de empresas, que utilizan las TIC con el objetivo de generar sistemas empresariales, altamente productivos y flexibles, con el propósito de mejorar su eficacia y desarrollar sus actividades comerciales. La disminución de los costes de los equipos, junto con los beneficios potenciales que se estima puede generar, está llamando a una inversión creciente en Tecnologías de Información y Comunicaciones.

Las TIC son incuestionables y están ahí, forman parte de la cultura tecnológica que rodea a la humanidad y con la que se debe convivir. Amplían las capacidades físicas y mentales y las posibilidades de desarrollo social[2].

El impacto económico y social que las TIC tienen en la sociedad hace que esté de moda el digitalizar casi todos los procesos de empresas, oficinas y demás sectores. Tal es el caso del Jardín Botánico "Oasis" que trata de incorporarse a las tecnologías de la información y las comunicaciones.

Luego de quedar vinculado con la Universidad de Cienfuegos, más específicamente con el Departamento de Agronomía de la Facultad de Mecánica, los especialistas relacionados se ponen en la tarea de hacer una compilación de las plantas que existen en el jardín.

Surge entonces la idea de crear un sistema para llevar el control de las especies que allí se conservan ya que actualmente no hay ningún procedimiento para tener control específico de lo que existe en el jardín, tan solo apuntes hechos por el dueño del lugar, quien, como único trabajador en el jardín, no dispone de tiempo para llevar una estadística de lo que posee y a su vez, atender todo el trabajo manual como el riego, arreglo, limpieza, etc. Se conformó una hoja de cálculo Excel pero teniendo en cuenta que no poseía la contundencia suficiente ni la facilidad que una herramienta directamente dirigida a los objetivos del jardín se decidió proceder a la confección de un software diseñado a objetivos específicos.

Expuesta la anterior situación problemática, se requiere entonces Elaborar un sistema informático que permita controlar las especies y familias que existen en el jardín botánico Oasis.

De este objetivo general se definen los objetivos específicos:

- Analizar la información de interés, referente a las diferentes especies que existen en el jardín botánico.
- Exportación e importación de ficheros de otros softwares.
- Diseño e implementación de una base de datos para almacenar toda la información relacionada con las especies botánicas del jardín.
- Diseño de la interfaz gráfica para la aplicación.
- Definición de la estructura de los datos y la información importante de las especies.
- Crear la ayuda y el manual de usuario del software.

Para el cumplimiento de estos objetivos se llevaran a cabo las siguientes tareas:

- Entrevista con la encargada del Jardín Botánico que trabajaría con los formularios de las especies y las familias botánicas.
- Recopilación de información y recomendaciones.
- Investigación de softwares existentes relacionados con el tema.

- Seleccionar el lenguaje de programación y el gestor de bases de dato para desarrollar el sistema.

Se define como idea a defender:

La utilización de un sistema informático permitirá un mejor control sobre la información de las especies y familias botánicas y a su vez brindará un balance del jardín en cuanto a índices botánicos se refiere.

Este sistema permitirá un control mas detallado sobre las especies y familias botánicas permitiendo la actualización de las mismas de una manera sencilla, así como brindará información de interés botánico al usuario, o sea los especialistas que atienden el jardín de forma automática y permitirá la ubicación de las especies en un mapa del jardín.

Para guiar el desarrollo de este trabajo se siguieron las pautas propuestas por el Proceso de Unificado de Desarrollo (RUP) y se utilizó el Lenguaje Unificado de Modelación (UML) para representar los artefactos que propone RUP en cada etapa del ciclo de vida del sistema propuesto.

El presente documento, estructurado en 4 capítulos, brindara la información de la siguiente manera:

Capítulo 1: Marco Teórico. En este capítulo se realizará un análisis sobre la problemática planteada, los conceptos necesarios para la comprensión del problema, un estudio sobre el estado del arte y también se analizarán las tecnologías existentes.

Capítulo 2: Modelo del Dominio y Requisitos: En este capítulo se expondrá el modelo del dominio, los conceptos del dominio y las reglas del negocio. También se abordará lo relacionado con el modelo del sistema, los casos de uso del sistema, los requerimientos funcionales y no funcionales.

Capítulo 3: Construcción de la solución propuesta. En este capítulo se explicará la solución propuesta a través del diagrama de clases de diseño, el modelo lógico de datos y el modelo físico de datos, así como el diagrama de implementación. Se describirá también los principios de diseño.

Capítulo 4: Estudio de factibilidad. En este capítulo se hará un estudio de la factibilidad de la construcción del sistema. Se utilizará el método de puntos de función. Además se describirán los beneficios tangibles e intangibles.

Capítulo I: Marco Teórico

Introducción

En este capítulo se mencionan las características del Jardín Botánico “Oasis”, se explican los principales conceptos relacionados con los jardines botánicos, se describe el estado del arte y se valorara las aplicaciones existentes relacionadas con el problema. Se hace un estudio de las tecnologías existentes para llegar a una conclusión de cual escoger teniendo en cuenta los requerimientos del problema para darle solución a éste. Se hace una descripción de la metodología a utilizar.

Descripción del dominio del problema

Botánica

Botánica, rama de la biología dedicada al estudio de las plantas (reino Plantae) y al de algunas otras clases de organismos como los hongos (reino Fungi). En la actualidad, las plantas se definen como organismos pluricelulares capaces de realizar la fotosíntesis. Pero otros organismos tradicionalmente llamados plantas, como las algas y los hongos, siguen formando parte de la botánica, por la relación histórica que mantienen con esta disciplina y por las muchas similitudes que hay entre ellos y las plantas verdaderas.

La botánica estudia todos los aspectos de las plantas, desde las formas más pequeñas y simples hasta las más grandes y complejas; y desde las características de los individuos aislados hasta las complejas interacciones de los distintos miembros de una comunidad botánica con su medio ambiente y con los animales.

Hoy por hoy los botánicos o especialistas en el estudio de las plantas desempeñan un abanico muy variado de actividades. Muchos ocupan puestos académicos y realizan labores de enseñanza e investigación, que comprenden trabajos de laboratorio y de campo. En términos estrictos, la botánica es una ciencia pura dedicada a estudiar la naturaleza básica de las plantas. Pero muchos aspectos de la botánica afectan directamente al bienestar y al progreso de la humanidad, por lo que la botánica aplicada ha cobrado gran importancia. Especialidades como la silvicultura y la horticultura están estrechamente vinculadas con la botánica básica; otras, como la farmacología y la agronomía, aunque son más autónomas, dependen también de los conocimientos botánicos básicos. Además, existen otras especialidades, como la geobotánica, que estudian la relación de las plantas con el medio físico.[3]

Jardín Botánico

Un jardín botánico es un jardín en el cual las plantas se cultivan y exhiben con fines primordialmente científicos y educativos. Un jardín botánico es una colección de plantas vivas mantenidas al aire libre o bajo cristal, en invernaderos. Muchos albergan también un herbario o colección de plantas secas, además de salas de lectura, laboratorios, bibliotecas, instalaciones museísticas y plantaciones experimentales o de investigación.

Las plantas pueden organizarse con arreglo a distintas categorías botánicas. Así, la organización puede ser sistemática (es decir, ajustada a la clasificación botánica de los ejemplares), ecológica (en función de su relación con el medio), o geográfica (por regiones de origen). Los jardines botánicos más grandes suelen contener además grupos especiales, como especies de roca, acuáticas, flores silvestres y colecciones de variedades cultivadas obtenidas mediante selección o mejora vegetal, como rosas, tulipanes o rododendros. Cuando la colección contiene sólo árboles, se habla de arboreto.[4]

Descripción del objeto de estudio

Descripción del entorno

El Jardín “Oasis” de José Manuel Aladro Roche, está situado en la finca “La Margarita”, en el Km. 8 de la carretera Cienfuegos-Rodas, en el Consejo Popular “Ariza”. Se funda en el año 1995, comenzando con una colección de Cactáceas y algunos helechos, pero su creador, estimulado por el interés despertado en los pobladores del lugar, continuó incrementando su colección, contando en la actualidad con unos 3000 ejemplares de más de 400 especies pertenecientes a 69 familias botánicas, cultivados en 2238 m² de terreno.

En el 2002 alcanza la categoría de Referencia Provincial otorgada por el Movimiento de la Agricultura Urbana que en el 2004 logra superarla y alcanza la Referencia Nacional, luego de un arduo trabajo en el 2006 alcanza la categoría de Excelencia Nacional que es la máxima que otorga este movimiento del país, la cual debe ser ratificada periódicamente en los recorridos que realiza el Grupo Nacional de Agricultura Urbana.

Este lugar se caracteriza por coleccionar plantas teniendo en cuenta principalmente su follaje, por lo que brinda al ojo humano un contraste de colores y formas muy diversas que lo convierten en un lugar ideal para el esparcimiento de los amantes de la naturaleza, distinguiéndose además por el empleo como tiestos o macetas, de los mas variados utensilios como llantas, ejes y tamboras de implementos y máquinas agrícolas fuera de servicio, recogidos por su creador en los lugares mas disímiles, lo que representa un ejemplo de cómo transformar los desechos en algo útil, bello y productivo, además de lograr con las propias plantas, barreras naturales y combinaciones de colores que permiten delimitar las áreas del patio.

En el patio, el productor ha ido creando condiciones de adaptación para las especies colectadas, de forma que hoy conviven exitosamente en el lugar, especies de requerimientos tan diferentes como las cactáceas, las begonias y la mariposa. Esto no

ha sido fácil de lograr implicando el movimiento constante de los tiestos por el patio, buscando suplir sus necesidades de iluminación con la sombra natural al no contar con un umbráculo para ello, así como un trabajo intenso de preparación de sustratos para cada especie colectada y manejo del riego para lograr satisfacer los requerimientos hídricos de los ejemplares plantados.

Es en el año 2008 que se vincula a la Universidad de Cienfuegos para brindar mantenimiento y ayuda experta de los especialistas de la Universidad ya que su propietario trabaja solo y esto dificulta la atención al jardín debido a su avanzada edad.

Herramientas Existentes

En la búsqueda de información sobre softwares existentes se encontró información de un software realizado por el Jardín Botánico Nacional llamado SISPLANT con fecha de terminación en 1996, hecho para difundirlo entre la red nacional de jardines pero nunca llegó a tal difusión. También se encontró el CollMan, un software para el manejo de especies vivas, hecho en conjunto el Jardín Botánico Nacional con el de Pinar del Río en el lenguaje Pascal para toda la red pero también quedó aislado en el Nacional. Estos dos softwares se presentan como catálogos de jardines y no poseen la automatización de índices importantes para cualquier botánico como lo son la riqueza específica o la dominancia por citar algunos, ni tampoco permite la ubicación en un mapa del jardín de las distintas plantas.

También se encontraron muchos programas multimedia y no de gestión de datos. Por ejemplo, en el Jardín Botánico de Cienfuegos existe la multimedia homónima soportada en formato html de carácter estático, o sea que no controla si murió alguna planta o si entró una nueva, así como que no realiza ningún cálculo estadístico.

Descripción del sistema actual

Actualmente sólo existe una hoja Excel que se confeccionó en el inicio de la cooperación con la Universidad donde se tiene toda la información de las familias y las especies pero incompleta y muy incómoda para trabajar y sin ningún tipo de metodología para su confección y de confección muy reciente, por esto se asume que no existe un sistema actual a cabalidad.

Descripción del objeto de automatización

Los procesos a informatizar serían: dar de alta a una nueva especie, modificar valores de una especie, dar de baja a una especie, dar de alta a una nueva familia, modificar valores de una familia, dar de baja a una familia, crear informes con los índices que sean de interés y la posibilidad de imprimirlos, la importación de archivos Excel y la exportación de archivos en formato XML, y la ubicación geográfica de los ejemplares de las especies.

Fundamentación de las metodologías a utilizar

UML

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje que permite especificar, visualizar, construir y documentar los artefactos de sistema de un software, así como modelado de negocio y de otros sistemas no softwares. El UML representa una colección de las mejores prácticas de ingeniería probadas satisfactoriamente en la modelación de grandes y complejos softwares[5].

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes.

Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos.

La definición del UML consiste en las partes siguientes:

- Semántica de UML. Se define la sintaxis semántica de UML. La arquitectura de UML está dividida por capas y está organizado por paquetes. Dentro de cada paquete hay elementos de modelado definidos en términos de sintaxis abstractas usando la notación de diagramas de clases de UML, reglas bien formadas usando texto y las expresiones del Lenguaje de Restricciones de Objetos, y las semánticas usando texto. Son incluidos dos apéndices: Glosario de UML y Elementos Estándares.
- Guía de Notación de UML. Define la notación y provee ejemplos de apoyo. La notación de UML representa la sintaxis gráfica para expresar la semántica descrita por metamodelos de UML.
- Extensiones de UML para Procesos de Objetos para Ingeniería del Software y para Modelado de Negocio. Estas extensiones incluyen aquellas para procesos y dominios específicos, en términos de mecanismos de extensiones e iconos de diagramas de procesos específicos.

Cuando se creó UML se tenían como objetivos primarios:

- Proveer a los usuarios un lenguaje de modelado visual que puedan desarrollar e intercambiar modelos.
- Proveer mecanismos de extensibilidad y especialización para extender su uso.
- Ser independiente de lenguajes de programación particulares y de procesos de desarrollo.
- Proveer bases formales para la comprensión del lenguaje de modelado.
- Fomentar el crecimiento del mercado de herramienta orientadas a objetos.

- Soportar conceptos de desarrollo de alto nivel como colaboraciones, frameworks, patrones, y componentes.
- Integrar las mejores prácticas.

UML es un lenguaje para especificar, crear, visualizar y documentar los artefactos de un sistema de software.

UML fusiona los conceptos de Booch, OMT y OOSE. El resultado es un lenguaje de modelado único, común, y ampliamente utilizable por los usuarios.

Proceso Unificado de desarrollo (RUP)

Muchas organizaciones lentamente se han dado cuenta de la importancia de un proceso de desarrollo de software bien-definido y bien-documentado para el éxito de sus proyectos de software. A través de los años, han recogido sus conocimientos y lo han compartido con otros desarrolladores. Este know-how (saber cómo) colectivo a menudo va mas allá de métodos, libros de textos, programas de entrenamiento, y pequeñas notas how-to acumuladas sobre algunos proyectos. Desafortunadamente estas prácticas a menudo van a parar en un polvoriento estante de algún desarrollador, volviéndose obsoletas rápidamente, y casi nunca continuadas.[6]. En contraste con el polvoriento estante, RUP es diseñado, desarrollado, distribuido, y mantenido como cualquier herramienta software.

El Proceso Unificado de Desarrollo, fue creado por el mismo grupo de expertos que crearon *UML*, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. Es un proceso de ingeniería de software. Provee de un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo era asegurar la producción de software de alta calidad que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos[6].

Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo (requisitos, análisis, diseño, implementación, prueba) que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software[7].

RUP captura muchas de las mejores prácticas en el desarrollo de software moderno en una forma que es apropiada para un amplio rango de proyectos y organizaciones.

Cubre las prácticas de desarrollo de software:

- Desarrollar software de forma iterativa
- Gestionar requerimientos
- Usar arquitecturas basadas en componentes
- Modelado visual del software
- Verificar la calidad del software continuamente
- Controlar los cambios al software

Fundamentación de las herramientas y lenguajes utilizados

Arquitectura de tres capas

La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. Las capas o niveles se separan en la de presentación, negocio y datos. Los términos capas y niveles no son exactamente lo mismo, ya que cuando se habla de capas se habla de separación de las capas desde el punto de vista lógico mientras que por niveles es cuando las capas están divididas de forma física.

En esta arquitectura se define de la siguiente manera:

- Capa de presentación: es la capa que ve el usuario donde se interactúa con éste en la muestra y recopilación de información. Sólo se comunica con la capa de negocio.
- Capa de negocio: es donde se encuentra la lógica del negocio, o sea, los requerimientos funcionales del software. El usuario no debe interactuar con esta capa de forma directa, la capa de presentación transmite los pedidos del usuario y ésta los devuelve por la capa de presentación. También se comunica con la capa de datos, la que accede a la base de datos, para manipular éstos.
- Capa de datos: es donde residen los datos y es la que se encarga de acceder al motor de bases de datos, el usuario no debe comunicarse directamente con esta capa.

C++

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje multiparadigma.

Una particularidad del C++ es la posibilidad de redefinir los operadores (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales.

C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos trae, con lo que obliga a usar librerías de terceros.

Los programas en C++ consisten en piezas llamadas clases y funciones. Se puede programar cada pieza de estas que se necesite para crear el programa. Sin embargo, la mayoría de los programadores en C++ se aprovechan de las ventajas de la rica colección de clases y funciones de las Librerías Estándar de C++. Usando las clases y funciones de la Librería Estándar de C++ en vez de las propias puede mejorar el rendimiento del programa ya que éstas son escritas cuidadosamente para aumentar la eficiencia, y también acorta el tiempo de desarrollo del programa. Pero realmente hay dos partes en el “mundo” de C++; una es saber el lenguaje en sí, y la otra es saber utilizar todas estas funciones y clases de la librería.[8]

C++ tiene ciertas características con respecto a otros lenguajes de programación. Los más notables son:

De programación orientado a objetos

La posibilidad de orientar la programación a los objetos permite al programador el diseño de aplicaciones desde un punto de vista como comunicación entre los objetos más que en una estructurada secuencia de código. Además, permite una mayor reutilización de código que es más lógico y productivo.

Portabilidad

Usted puede compilar prácticamente el mismo código C++ en casi cualquier tipo de ordenador y sistema operativo sin realizar ningún cambio. C++ es el lenguaje de programación más utilizado y portado en el mundo.

Brevedad

El código escrito en C++ es muy corto en comparación con otros idiomas, ya que el uso de caracteres especiales se prefiere a las palabras clave, algunas de ahorro de esfuerzo para el programador (¡y la prolongación de la vida de nuestros teclados!).

Programación modular

El cuerpo de una aplicación en C++ puede estar compuesto de varios archivos de código fuente que se compilan por separado y después juntos vinculados. Ahorrar tiempo, ya que no es necesario recompilar la aplicación completa al realizar un solo cambio, pero sólo el archivo que lo contiene. Además, esta característica permite vincular código C++ con el código producido en otros idiomas, como el ensamblador o C.

Compatibilidad C

C++ es compatible con el lenguaje C. Cualquier código escrito en C puede ser fácilmente incluido en un programa C++ sin hacer ningún cambio.

Velocidad

El resultado de un código C++ en compilación es muy eficiente, debido a su dualidad como de alto nivel y de bajo nivel de idiomas y la reducción del tamaño de la lengua propia.

Borland C++ Builder 6

C++Builder 6 aumenta la fuerza de la aclamada herramienta Borland's ANSI C++ del tipo RAD (Rapid Application Development o aplicación de rápido desarrollo) con abundantes nuevas características, incluyendo soportes para aplicaciones, Web y múltiples tecnologías de base de datos tecnologías.

Entre las características se tienen:

BizSnap

- Creación de aplicaciones clientes para Web Services existentes.

- Creación de Web Services estándares y su publicación usando WSDL.
- Creación de aplicaciones que trabajan con XML, conversión de documentos XML a tipos nativos, y transformar documentos XML a datasets.

WebSnap

- Los diseñadores de interfaces de páginas Web permiten de manera rápida crear y ver aplicaciones WebSnap en HTML con vistas preliminares actualizadas instantáneamente en el navegador.
- Programación del lado del servidor (Server-side scripting) permite interactuar fácilmente con objetos WebSnap/NetCLX.
- Los componentes web pre-construidos incluyen datasets, formularios de autenticación y otros elementos comúnmente usados en aplicaciones web.
- Nuevos asistentes que facilitan la creación de aplicaciones con módulos que incluyen los componentes que se necesitan.
- C++ Builder incluye cambios a las características de WebBroker para soportar el servidor web Apache, para encajar bien en la arquitectura WebSnap y para hacer más fácil el debugueo de aplicaciones web.

DataSnap

- Provee soporte para aplicaciones con bases de datos multi-tier usando SOAP para comunicarse entre los clientes y el servidor.

Soporte para bases de datos

- Incluye muchas nuevas características para bases de datos tales como el nuevo mecanismo de acceso a datos llamado dbExpress (conjunto de drivers para manejar servidores SQL como Oracle, etc.) y nuevos componentes.

Nueva Librería Estándar C++ STLport

- C++ Builder incluye la implementación de la Librería Estándar C++ de STLport y usa esta librería por defecto.

Librería de componentes multi-plataforma

- C++ Builder 6 viene con la Librería de Componentes Borland para multi-plataforma (CLX), una librería de clases muy similar a la VCL que está diseñada para correr tanto en Windows como en Linux. La mayoría de los nombres de los objetos de la CLX son iguales a los de la VCL así también como las propiedades, métodos y eventos.

Sistema de ayuda extensivo

C++ Builder permite hacer llamadas a distintos visores de ayudas mas allá de los visores estándares de Windows permitiendo así crear ayudas tanto en Windows como en Linux.

- Nuevas interfaces que comunican la aplicación con el visor de ayuda. Las interfaces están definidas en HelpIntfs.hpp e incluyen clases como IExtendedViewer, ISpecialHelpWinViewer, IHelpManager, IHelpSystem, IHelpSelector, e ICustomHelpViewer.
- El administrador de ayudas mantiene una lista de las ayudas asociadas y hace las llamadas.
- La VCL provee una implementación de ICustomHelpViewer diseñada para comunicarse con WinHelp.

Soporte para CORBA

- Incluye VisiBroker 4.5, con soporte para desarrollo de clientes y servidores CORBA.
- Incluye Borland AppServer v4.5 SIDL Support, permitiendo usar el C++ Builder para crear aplicaciones Web Services que usan EJB para AppServer.

Características del IDE

El IDE se ha enfocado a mejorar la productividad y proveer fácil acceso a todos los nuevos componentes.

- El diseñador para módulos de datos tiene elementos disponibles aun cuando no se usa ningún data module.

- El visor de árboles de objetos muestra las relaciones lógicas entre los componentes del proyecto.
- El inspector de objetos cuenta con opciones tales como referencias a las propiedades del objeto seleccionado en la lista que trae consigo. Una página inspector de objetos en opciones de entorno lo hace más configurable.
- En opciones de entorno hay varias páginas de configuración como la de variables de entorno.
- La herramienta CodeInsight permite el completamiento de código y de parámetros.

Algunas de estas características no están disponibles en todas las ediciones del C++ Builder, puesto que hay tres: edición empresarial, profesional y estándar.

Las ediciones tienen diferentes precios como lo muestra la siguiente lista:

| | |
|--|------------|
| C++ Builder 6 Ingles Personal | 76,50 € |
| C++ Builder 6 Ingles Professional - Nueva licencia | 1.099,00 € |
| C++ Builder 6 Ingles Professional - Actualización | 449,00 € |
| C++ Builder 6 Ingles Enterprise - Nueva licencia | 3.389,00 € |
| C++ Builder 6 Ingles Enterprise - Actualización | 2.279,66 € |
| C++ Builder 6 Professional Educational Edition | 118,80 € |
| C++ Builder 6 Enterprise Educational Edition | 430,65 € |

Java

JAVA es un lenguaje de programación orientado a objetos desarrollado por la empresa Sun Microsystems en 1995 y que se ha extendido ampliamente en World Wide Web. Es un lenguaje de alto nivel y propósito general similar a C++, con marcadas características de seguridad y transportabilidad. Este lenguaje define una máquina virtual independiente de la plataforma donde se ejecuta, que procesa programas, llamados Applets, descargados desde el servidor Web. Además, debido al modo de ejecución de los Applets, este lenguaje es muy seguro frente a la presencia y ataque de virus informáticos.[9]

JAVA es un lenguaje de programación de propósito general que puede ser empleado para desarrollar cualquier tipo de aplicación principalmente aquellas referidas a Internet.

Java es pequeño, simple, seguro, Orientado a objetos, produce código byte neutral de la arquitectura, posee un colector de basura, es un lenguaje de programación multitarea con un mecanismo de manejo de excepciones para la escritura distribuida.[10]

La plataforma Java está compuesta de dos componentes esenciales: la máquina virtual de Java y las bibliotecas estándar Java Application Programming Interface (Java API). Las librerías de clases simplifican la programación en Java y aceleran el desarrollo comparado con otros lenguajes.

De acuerdo con la definición de Java que da Sun Microsystems, los atributos de Java son:

- Simple. Los desarrolladores de Java quitaron muchas de las características innecesarias de otros lenguajes de programación de alto nivel. Por ejemplo, Java no soporta punteros, casteo de tipos, estructuras o uniones, sobrecarga de operadores, plantillas, ficheros de encabezado ni herencia múltiple.
- Orientado a objetos. Como C++, Java utiliza clases para organizar el código en módulos lógicos. En tiempo de ejecución, el programa crea objetos de esas clases. Las clases en Java pueden heredar de otras clases, pero herencia múltiple, cuando una clase hereda atributos y métodos de más de una clase, no está permitido.
- De tipo estático. Todos los objetos usados en un programa deben estar previamente declarados antes de ser usados. Esto permite al compilador de Java localizar y reportar conflictos entre tipos.
- Compilado. Antes de ejecutar un programa escrito en Java, el programa debe ser compilado. La compilación resulta en un fichero “byte-code” que, similar al fichero “machine-code”, puede ser ejecutado en cualquier sistema operativo que tenga un intérprete Java. Este intérprete lee el byte-code y lo traduce a lenguaje de

máquina que puede ser ejecutado directamente por la máquina que está ejecutando el programa en Java. Se podría decir que Java es un lenguaje tanto interpretado como compilado.

- Multi-threaded. Los programas en Java pueden tener varios hilos de ejecución, lo cual permite al programa manejar diferentes tareas al mismo tiempo. Por ejemplo, un programa multi-hilo puede renderizar una imagen en la pantalla en un hilo mientras acepta entradas del teclado por el usuario en el hilo principal. Toda aplicación posee al menos un hilo, el cual representa el camino de ejecución principal del programa.
- Recogedor de basura. Los programas Java recogen su propia basura, lo que significa que los programas no necesitan destruir los objetos existentes en memoria. Esto virtualmente libera a los programadores de los problemas de manejo de memoria.
- Robusto. Como el intérprete de Java chequea todos los accesos al sistema realizados dentro de un programa, los programas Java no pueden colapsar el sistema. Al contrario, cuando un error serio es encontrado, Java crea una excepción. Esta excepción puede ser capturada y manipulada por el programa sin riesgo de afectar al sistema.
- Seguro. Java no solo verifica todos los accesos a memoria sino que también asegura que ningún virus “coja botella” con una applet corriendo. Como los punteros no son soportados por Java, los programas no pueden acceder a áreas del sistema que no tienen autorización.
- Extensible. Java soporta métodos nativos, que son funciones escritas en otros lenguajes de programación, usualmente C++. Soporte para métodos nativos permite a los programadores escribir funciones que se ejecuten más rápido que su equivalente en Java. Los métodos nativos son vinculados dinámicamente al programa Java; lo que significa que son vinculados en tiempo de ejecución. Como el Java está refinado para aumentar velocidad, los métodos nativos probablemente sean innecesarios.
- De fácil entendimiento. Java está basado en tecnología que ha sido desarrollada por muchos años. Por esta razón Java puede ser entendido rápida y fácilmente

por cualquiera que tenga experiencia en lenguajes de programación modernos como el C++.

Java puede ser usado para crear dos tipos de programas: applets y aplicaciones stand-alone. Una applet es simplemente una parte de una página Web, como lo son una imagen o una línea de texto. Como mismo un navegador se encarga de mostrar una imagen referenciada en un documento HTML, un navegador con soporte para Java localiza y ejecuta una applet, cuando el navegador Web con Java carga el documento HTML, también carga y ejecuta la applet.

Usando applets se puede hacer todo desde añadir un gráfico animado a una página Web hasta crear juegos completos y herramientas que pueden ser ejecutados sobre Internet.

Aunque la mayoría de los programadores de Java están emocionados por la habilidad de crear applets, en Java también se pueden crear aplicaciones stand-alone, que son aplicaciones que no necesitan estar embebidas en documentos HTML. La aplicación mas conocida es el navegador Web HotJava en sí. Este navegador está escrito completamente en Java, demostrando como Java maneja no solo tareas normales como ciclos y expresiones matemáticas, sino también como puede manejar las complejidades de la programación de telecomunicaciones.[11]

SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por la mayoría de los servidores de base de datos relacionales actuales, ya que está estandarizado. SQL se utiliza para permitir a los usuarios de la base de datos por medio de un lenguaje semejante a inglés realizar consultas y de esa manera obtener los datos que necesiten. Está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.[12]

Fundamentación del gestor de bases de datos utilizado

Microsoft Office Access 2003

Microsoft Office Access es un sistema de gestión de bases de datos creado por Microsoft para uso personal o de pequeñas organizaciones. Es un componente de la suite Microsoft Office. Su función es ser un potente gestor de base de datos, capaz de trabajar en sí misma o bien con conexión hacia otros lenguajes de programación, tales como Visual Basic 6.0 o Visual Basic .NET. Pueden realizarse consultas directas a las tablas contenidas mediante instrucciones SQL.[13]

Entre las principales funcionalidades de Access se encuentran:

- Crear tablas de datos indexadas.
- Modificar tablas de datos.
- Relaciones entre tablas (creación de bases de datos relacionales).
- Creación de consultas y vistas.
- Consultas referencias cruzadas.
- Consultas de acción (INSERT, DELETE, UPDATE).
- Formularios.
- Informes.
- Interacción con otras aplicaciones que usen VBA (resto de aplicaciones de Microsoft Office, Autocad, etc.).
- Macros.

Uno de sus mayores inconvenientes es que no es multiplataforma, pues sólo está disponible para sistemas operativos de Microsoft. Su uso es inadecuado para grandes proyectos de software que requieren tiempos de respuesta rápido o muchos accesos simultáneos a la base de datos. También solo admite un número limitado de conexiones.

MySQL

MySQL es uno de los gestores de bases de datos más populares desarrollados bajo la filosofía de código abierto, puede utilizarse gratuitamente y su código fuente está disponible. Es desarrollado y mantenido por la empresa MySQL AB y considerado una de las aplicaciones de código abierto más exitosas, abriéndose camino gracias a su nivel competitivo y su accesibilidad. Entre los clientes de la compañía se cuentan nada menos que la NASA, Yahoo, Suzuki, Google y la Agencia Associated Press[14].

MySQL es el servidor de base de datos predilecto por los desarrolladores de páginas web dinámicas debido a las siguientes características:

- Alto rendimiento
- Alta velocidad al servir consultas
- Recuperación de errores, respaldo, etc.
- Robusto
- Excelente integración con PHP
- No tiene límites de tamaños de tablas en cuanto a registros
- Control de acceso de usuarios, permisos de acceso a tablas
- Es multi-hilo, maneja muchas conexiones al mismo tiempo sin que una tenga que esperar por otra.
- Multiplataforma

Conclusiones del capítulo

Después de haberse realizado un análisis teórico del problema se arribó a las siguientes conclusiones:

El sistema se construirá utilizando el lenguaje de programación orientado a objetos C++ con la herramienta C++ Builder 6 y como motor de bases de datos se utilizará Microsoft Office Access 2003 debido a que el desarrollador tiene mayor dominio sobre el lenguaje C++, el usuario que utilizará el producto trabajará en la plataforma Windows, el gestor

de bases de datos Access posee todas las características necesarias para la elaboración del sistema además de no ser potente pero si seguro y se dispone de poco tiempo para el desarrollo del sistema propuesto. El sistema contará con los módulos necesarios para el trabajo con la información de las especies existentes en el Jardín. Para el modelado del proceso de desarrollo de software se utilizará el Lenguaje de Modelado Unificado siguiendo la metodología RUP (Proceso Unificado de Desarrollo), ya que cumple con los cuatro objetivos o roles que debe tener: proveer una guía para el orden de las actividades del equipo, especificar cuales artefactos deben ser desarrollados y cuando deberían ser desarrollados, dirigir las tareas de los desarrolladores y las del equipo como un todo, y ofrecer criterios para monitorear y medir los productos y las actividades del proyecto[6].

Capítulo 2 Modelo del Dominio y Requisitos

Introducción

En este capítulo se tratarán los conceptos asociados al dominio del problema así como las relaciones entre éstos y se precisarán las reglas de negocio implicadas en este trabajo; también se hará una descripción del sistema propuesto. Se utilizarán los artefactos de RUP modelo del dominio y modelo de casos de usos del sistema para una mayor comprensión del sistema tanto de sus entidades y conceptos como de sus requerimientos y su solución propuesta.

Definición de las entidades y los conceptos principales

Especies botánicas: Cada uno de los grupos en que se dividen los géneros y que se componen de individuos que, además de los caracteres genéricos, tienen en común otros caracteres por los cuales se asemejan entre sí y se distinguen de los de las demás especies. La especie se subdivide a veces en variedades o razas[15].

Familias botánicas: Taxón constituido por varios géneros naturales que poseen gran número de caracteres comunes[16].

Índices botánicos: Mediciones que indican el estado total o parcial de un ecosistema en cuanto a especies botánicas se refiere.

Reglas del negocio a considerar

Las reglas de negocio describen las principales políticas que deben cumplirse para el adecuado funcionamiento del negocio. Para llevar a cabo el trabajo con las especies en el sistema es necesario tener en cuenta las siguientes reglas.

- En el sistema pueden haber tantas familias y especies como hayan en el jardín.
- Las especies solo pueden pertenecer a una sola familia y una familia puede tener cuantas especies tenga definidas (internacionalmente).
- Las especies pueden tener ninguna o tantas ubicaciones como cantidad de ejemplares tenga el jardín.
-

Representación del modelo del dominio

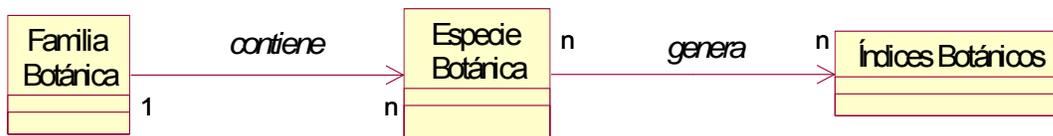


Figura 1 Modelo de Dominio

Descripción del sistema propuesto

Concepción general del sistema

El sistema consiste en el ejecutable SCEFBJO.exe y la base de datos dboasis.mdb. El ejecutable SCEFBJO.exe se obtuvo en el Borland C++ Builder 6, y la base de datos fue realizada en el Microsoft Office Access 2003. El sistema propuesto ofrece un ambiente agradable de fácil manejo para garantizar un adecuado proceder en el sistema.

El sistema en su ventana principal ofrece todas las opciones a las que se pueden acceder:

- Familias botánicas
- Especies botánicas
- Reportes
- Conversión de datos
- Mapa

Cada una de estas opciones lleva a una ventana más específica que facilitará el trabajo del especialista que se encuentre utilizando el sistema.

En el caso de las familias botánicas conduce a la ventana principal de las familias botánicas donde se muestra una lista de todas las familias existentes en el sistema, teniendo un menú con las tareas relacionadas a las familias botánicas como lo son añadir una familia, modificar una familia y eliminar una familia.

Si la opción es especies botánicas entonces va a la ventana principal de las especies donde se muestra una lista de las especies guardadas en la base de datos dboasis, en la lista aparecerá el nombre científico, el nombre vernáculo y la familia a la que pertenece. Cuenta con un menú de tareas relacionadas con las especies botánicas como lo son añadir especie, eliminar especie y mostrar la especie en el mapa. En el caso de la última tarea se abre una ventana con un mapa del jardín y sus especies ubicadas, en el mapa se puede añadir o eliminar ubicaciones, se puede filtrar por especie para ver solamente las ubicaciones de una, además se puede agrandar el mapa permitiendo al especialista una mejor ubicación en el mapa. También permite asignarle una fotografía a la especie para darle más detalle a la información de las especies cuando se quiera ver sus características.

Para los reportes se debe escoger el tipo de reporte y en caso de que lleve algún parámetro el especialista puede entrarlo de una manera sencilla. Al escoger el reporte se muestra el reporte en vista preliminar con las opciones de guardar el reporte (en formato quickreport) e imprimirlo con las pertinentes configuraciones de impresora.

El sistema permite la importación y exportación de la base de datos a dos soportes: XML y Excel. Con el objetivo de si se quiere guardar el estado del jardín en una fecha cualquiera se implementó la exportación de las tablas, y si se quiere revisar este estado solo se tiene que importar el fichero que se creó con anterioridad.

Requerimientos funcionales

Los requerimientos funcionales son un conjunto de funcionalidades que son impuestas por el usuario para que el sistema cumpla con sus objetivos. El objetivo de los requerimientos es describir que debe hacer el sistema[17].

Para este sistema se definieron los siguientes requerimientos funcionales:

- Insertar familia botánica
- Modificar familia botánica
- Listar familia botánica
- Buscar familia botánica
- Eliminar familia botánica
- Insertar especie botánica
- Modificar especie botánica
- Listar especie botánica
- Mapear especies botánicas
- Buscar especie botánica por nombre científico
- Buscar especie botánica por nombre vernáculo
- Eliminar especie botánica
- Exportar base de datos a Excel
- Importar base de datos desde Excel
- Exportar base de datos a XML
- Importar base de datos desde XML
- Calcular índices botánicos
- Mostrar índices botánicos
- Imprimir índices botánicos

- Mostrar reporte de especies con cantidad de ejemplares mayor de x
- Imprimir reporte de especies con cantidad de ejemplares mayor de x
- Mostrar reporte de especies dada la familia
- Imprimir reporte de especies dada la familia
- Mostrar reporte de todas las familias con sus especies
- Imprimir reporte de todas las familias con sus especies

Requerimientos no funcionales

Apariencia o interfaz externa. La interfaz estará basada en el estándar de ventanas de Windows. El sistema le ofrecerá al usuario la posibilidad del control total en todo momento, brindando navegabilidad de manera sencilla.

Usabilidad. El sistema está orientado a los especialistas botánicos de la Universidad de Cienfuegos encargados del jardín Oasis. El usuario del sistema no necesitará de grandes conocimientos informáticos para la explotación del producto.

Rendimiento. Los tiempos de respuesta del sistema son adecuados al volumen de datos manipulados, estos serán generalmente instantáneos o de corta duración. Sólo los trabajos con ficheros Excel pueden demorar más en dependencia de las características de la maquina y la cantidad de información a trabajar.

Soporte. El sistema dispondrá de una arquitectura de diseño flexible que facilite su perfeccionamiento gradual, así como la incorporación de funcionalidades que aumenten las potencialidades del mismo.

Portabilidad. El sistema debe ser desarrollado sobre la plataforma Windows.

Requerimientos Legales. Este software es propiedad intelectual de la Universidad de Cienfuegos y solo es permitida su comercialización con el consentimiento de los autores y la entidad involucrada.

Confiabilidad. El sistema debe presentar un mecanismo de respuesta rápida ante fallos y que en caso de ocurrencia se minimicen las pérdidas de información.

Ayuda y documentación en línea. Debe disponerse de una ayuda bien detallada sobre las principales opciones del sistema.

Software. Se tiene que disponer del sistema operativo Windows 2000 o mayor y tener instalado el Microsoft Office o al menos el driver para bases de datos Access.

Hardware. Se debe contar con al menos:

- Procesador Pentium
- 128 Mb. de RAM
- 1 Gb. de HDD Libre
- Mouse
- Teclado

Restricciones en el diseño y la implementación. Para garantizar una mejor documentación del sistema, se utiliza en el análisis, diseño e implementación del sistema propuesto el UML (Unified Modeling Language) y como herramienta de apoyo a este lenguaje de modelación se hace uso del Rational Rose.

Modelo de casos de uso del sistema

Un modelo de casos de uso es un modelo que describe los requerimientos funcionales del sistema u otro clasificador en términos de casos de uso. El modelo de casos de uso representa la funcionalidad de un sistema u otro clasificador como interacto externo manifiesto con el sistema[18].

El modelado por casos de uso es un tipo especializado de modelado estructurado interesado en modelar la funcionalidad de un sistema. Usualmente se usa modelado por casos de uso durante las actividades de requerimientos para capturar los requerimientos que definen lo que el sistema debe hacer. El modelado por casos de uso

típicamente comienza temprano en un proyecto y continúa a través del proceso de desarrollo del sistema. Usualmente se hace como una serie de entrevistas entre los usuarios y los analistas del sistema en las cuales pueden explorar ideas y los requerimientos pueden conocerse a tiempo[19].

Actores del sistema

Un actor es un usuario o un sistema externo con el cual en sistema modelado interactúa. Un actor es externo al sistema, interactúa con el, puede ser un humano u otro sistema, y tiene objetivos y responsabilidades a satisfacer al interactuar con el sistema. Actores trata la pregunta de quién y qué interactúa con el sistema[19].

| Actor | Descripción |
|-----------------------|--|
| Especialista botánico | Realiza todas las acciones en el sistema teniendo acceso a todos los requerimientos funcionales del sistema. |

Diagramas de casos de uso del sistema

Un diagrama de casos de uso es un diagrama que muestra la relación entre actores y casos de uso en un sistema. Un gráfico que muestra actores, un conjunto de casos de usos, asociaciones entre actores y casos de uso, relaciones entre casos de uso y generalizaciones entre actores. Los diagramas de casos de uso muestran elementos del modelo de casos de uso[18].

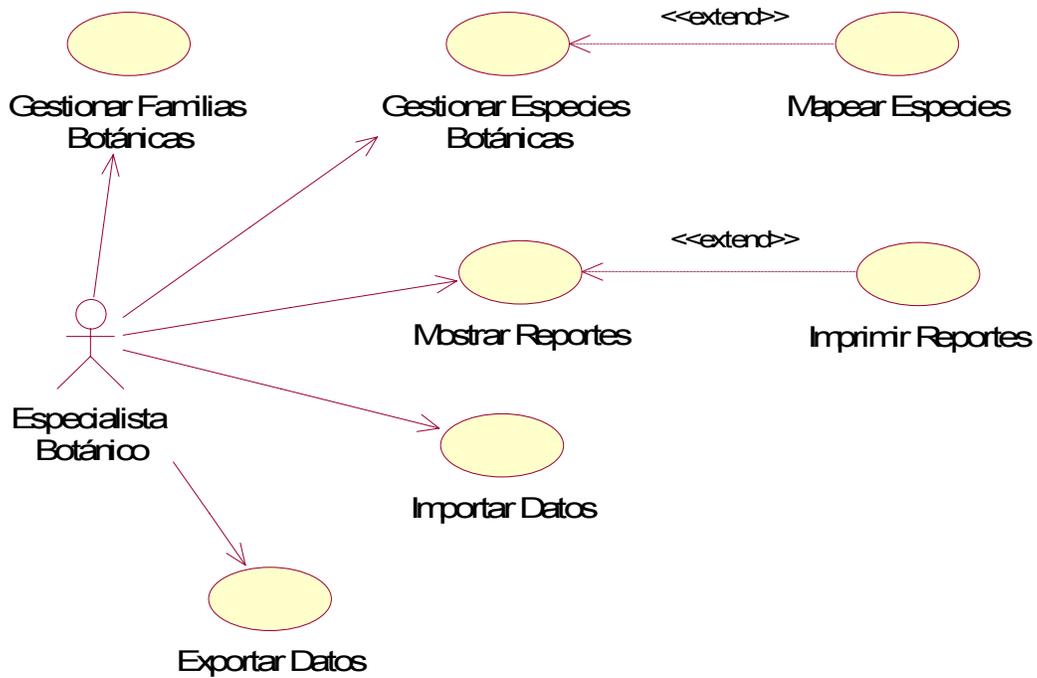


Figura 2 Diagrama de casos de uso del sistema

Descripción de los casos de uso del sistema

| | |
|--------------------|--|
| Caso de uso | Gestionar Familias Botánicas |
| Actor | Especialista botánico(inicia) |
| Propósito | Permitir insertar, eliminar o modificar los datos de las familias botánicas. |
| Resumen | El caso de uso se inicia cuando el especialista desea insertar, modificar, ver los detalles de alguna, o eliminar una familia botánica. |
| Referencias | Insertar familia botánica Modificar familia botánica Listar familia botánica Buscar familia botánica Eliminar familia botánica |

| | |
|------------------------------|--|
| Precondiciones | - |
| Post-condiciones | Se actualiza la información si acción: <ul style="list-style-type: none"> - Inserta: se guarda los datos de la familia nueva. - Modifica: se guardan los nuevos datos para la familia indicada. - Elimina: se elimina la familia de la base de datos. |
| Requisitos Especiales | - |
| Prototipo | Ver anexo Figuras 8, 9, 10 |

Tabla 1 Descripción del caso de uso Gestionar Familias Botánicas

| | |
|-------------------------|--|
| Caso de uso | Gestionar Especies Botánicas |
| Actor | Especialista botánico(inicia) |
| Propósito | Permitir insertar, eliminar o modificar los datos de las especies botánicas. |
| Referencias | Insertar especie botánica Modificar especie botánica Listar especie botánica Buscar especie botánica por nombre científico Buscar especie botánica por nombre vernáculo Eliminar especie botánica |
| Resumen | El caso de uso se inicia cuando el especialista desea insertar, modificar, ver los detalles de alguna, o eliminar una especie botánica. |
| Precondiciones | - |
| Post-condiciones | Se actualiza la información si acción: <ul style="list-style-type: none"> - Inserta: se guarda los datos de la especie nueva. - Modifica: se guardan los nuevos datos para |

| | |
|------------------------------|---|
| | <p>la especie indicada.</p> <ul style="list-style-type: none"> - Elimina: se elimina la especie de la base de datos. |
| Requisitos Especiales | - |
| Prototipo | Ver anexo Figuras 11, 12, 13, 14 |

Tabla 2 Descripción del caso de uso Gestionar Especies Botánicas

| | |
|------------------------------|---|
| Caso de uso | Mostrar Reportes |
| Actor | Especialista botánico(inicia) |
| Propósito | Brindar diferentes reportes de interés al especialista botánico. |
| Referencias | <p>Calcular índices botánicos</p> <p>Mostrar reporte de especies con cantidad de ejemplares mayor de x</p> <p>Mostrar reporte de especies dada la familia</p> <p>Mostrar reporte de todas las familias con sus especies</p> |
| Resumen | <p>El caso de uso se inicia cuando el especialista desea ver algún reporte determinado.</p> |
| Precondiciones | Se debe escoger el tipo de reporte que se desea ver. |
| Post-condiciones | - |
| Requisitos Especiales | - |
| Prototipo | Ver anexo Figura 15 |

Tabla 3 Descripción del caso de uso Mostrar Reportes

| | |
|--------------------|--------------------------------|
| Caso de uso | Imprimir Reportes |
| Actor | Especialista botánico(inicia) |

| | |
|---|--|
| Propósito | Imprimir el reporte seleccionado. |
| Referencias | Imprimir reporte de especies dada la familia Imprimir reporte de especies con cantidad de ejemplares mayor de x Imprimir índices botánicos Imprimir reporte de todas las familias con sus especies Caso de uso Crear Reportes (extend) |
| Resumen El caso de uso se inicia cuando el especialista desea imprimir el reporte activo. | |
| Precondiciones | Debe haber un tipo de reporte activo. |
| Post-condiciones | Se imprime el reporte. |
| Requisitos Especiales | - |
| Prototipo | - |

Tabla 4 Descripción del caso de uso Imprimir Reportes

| | |
|--|--|
| Caso de uso | Importar Datos |
| Actor | Especialista botánico(inicia) |
| Propósito | Permitir la importación de ficheros XML y Excel hacia la base de datos. |
| Referencias | Importar base de datos desde Excel Importar base de datos desde XML |
| Resumen El caso de uso se inicia cuando el especialista desea importar datos desde un fichero XML o Excel. | |
| Precondiciones | Para importar ficheros éstos deben estar en el formato especificado en la ayuda. |
| Post-condiciones | Al importar se llena la base de datos con nuevos valores con la información existente en el fichero importado. |

| | |
|------------------------------|--|
| Requisitos Especiales | El Microsoft Office Excel debe estar instalado en la computadora a usar. |
| Prototipo | - |

Tabla 5 Descripción del caso de uso Importar Datos

| | |
|---|---|
| Caso de uso | Exportar Datos |
| Actor | Especialista botánico(inicia) |
| Propósito | Permitir la exportación de la base de datos hacia ficheros XML y Excel. |
| Referencias | Exportar base de datos a Excel Exportar base de datos a XML |
| Resumen El caso de uso se inicia si se desea guardar la base de datos en XML o Excel. | |
| Precondiciones | La base de datos debe tener algún registro para poder exportarla. |
| Post-condiciones | Al exportar la base de datos se crean nuevos ficheros del tipo especificado con la información requerida. |
| Requisitos Especiales | El Microsoft Office Excel debe estar instalado en la computadora a usar. |
| Prototipo | - |

Tabla 6 Descripción del caso de uso Exportar Datos

| | |
|--|---|
| Caso de uso | Mapear Especies |
| Actor | Especialista botánico(inicia) |
| Propósito | Permitir la ubicación geográfica de algunos o todos los ejemplares de las especies botánicas existentes en el jardín. |
| Referencias | Mapear especies botánicas |
| Resumen El caso de uso se inicia cuando el especialista desea ubicar geográficamente | |

| | |
|---|--|
| en un mapa del jardín los ejemplares de las especies. | |
| Precondiciones | La especie debe existir en la base de datos. |
| Post-condiciones | - |
| Requisitos Especiales | - |
| Prototipo | Ver anexo Figura 16 |

Tabla 7 Descripción del caso de uso Mapear Especies

Conclusiones

En este capítulo se explicaron los principales conceptos relacionados con el dominio y sus relaciones, se definieron las reglas del negocio, y además se definieron los requerimientos funcionales y no funcionales. Usando el artefacto modelo de casos de uso del sistema de la metodología RUP se logró definir la forma en que los actores interactúan con los casos de usos, y se definieron éstos.

Capítulo 3 – Construcción de la solución propuesta

Introducción

En este capítulo se describe la construcción de la solución propuesta. Se utiliza de la metodología Proceso Unificado de Desarrollo los artefactos diagrama de clases de diseño, modelo lógico de datos, y el modelo físico de datos para la descripción de la solución propuesta. También se describirán los estándares de

Diagrama de clases del diseño

El diagrama de clases es una presentación gráfica que muestra una colección de elementos de modelo estáticos tales como clases, tipos, y sus contenidos y relaciones. Un diagrama de clases puede mostrar una vista de un paquete y puede contener símbolos para paquetes anidados[18].

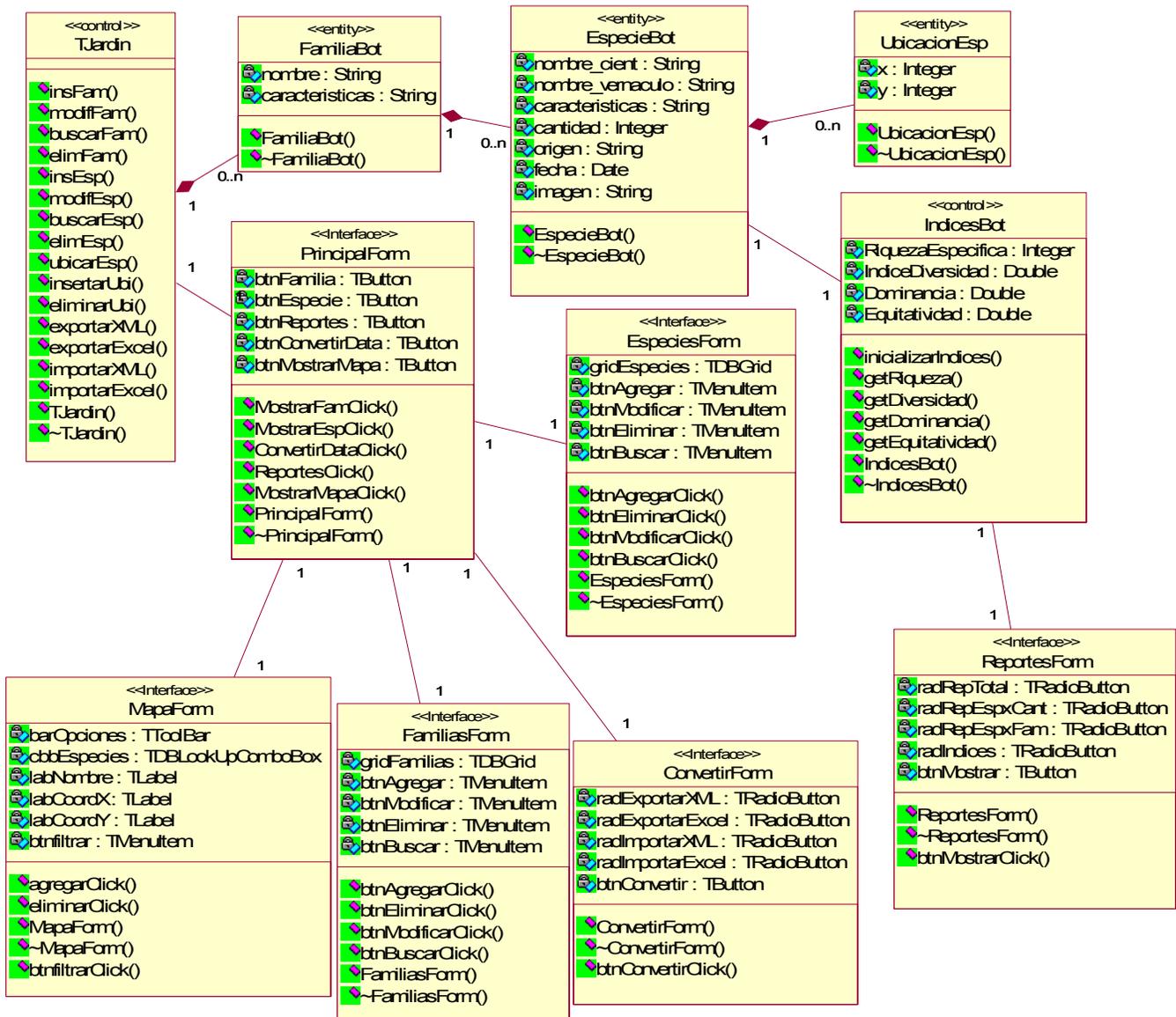


Figura 3 Diagrama de clases de diseño

Diseño de la base de datos

Modelo lógico de datos

El modelo lógico de la base de datos determina cómo se estructuran los datos de forma lógica mediante tablas y relaciones. Este diseño puede tener también una gran repercusión en el rendimiento de la aplicación[20].

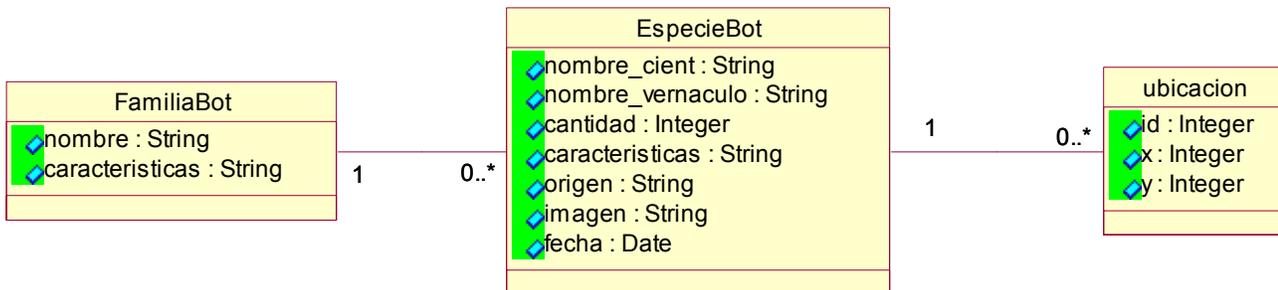


Figura 4 Modelo lógico de datos

Modelo físico de datos

El modelo físico de datos incluye todos los aspectos de diseño de un modelo de base de datos que se pueden modificar sin cambiar los componentes de la aplicación[20].

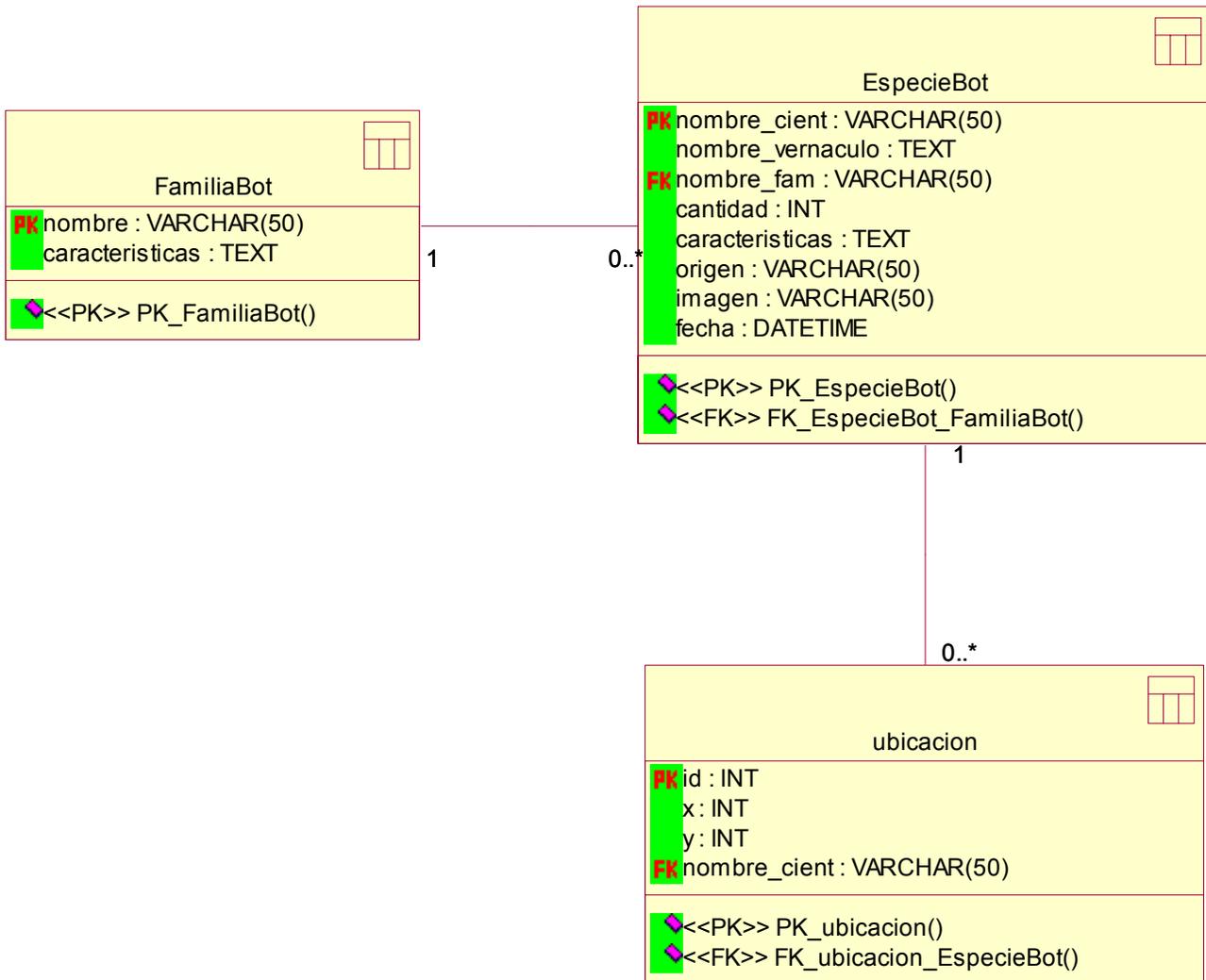


Figura 5 Diagrama del Modelo físico de datos

Diagrama de implementación

El modelo de implementación describe la forma en que los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. Describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros[13].

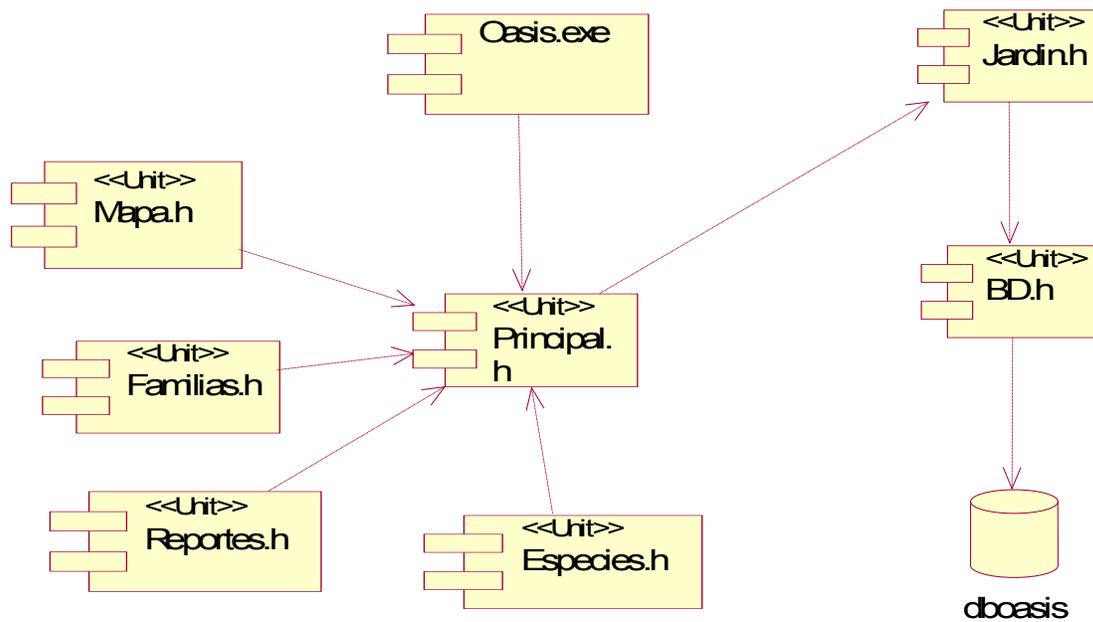


Figura 6 Diagrama de Implementación

Principios de diseño

Estándares en la interfaz de la aplicación

La interfaz diseñada está basada en el estándar de ventanas de Windows. El tipo de letra utilizado es Times New Roman tamaño 8. La carga visual es adecuada. El sistema está conformado por módulos que parten de una ventana principal, estos módulos se comunican entre si, y cada uno posee una barra de herramientas con las tareas específicas del módulo para ganar en claridad y separación de los temas por así llamarlos.

Tratamiento de errores

El diseño de la interfaz está dirigido a evitar errores. Los mensajes de errores que emite el sistema se muestran de forma concisa, explicando el por qué del mensaje así como la sugerencia para evitarlo. También los mensajes de advertencias sobre los cambios que no se pueden deshacer son explicativos y brindan las opciones de continuar o cancelar.

Concepción General de la ayuda

Adjunto con el sistema viene una ayuda que brinda al usuario una detallada explicación de cómo sería la manipulación del mismo. La ayuda se encuentra disponible en la ventana principal del sistema y en todos sus módulos con sólo presionar la tecla F1. La ayuda está formada por un menú interno que contiene todas las funcionalidades del sistema. El usuario podrá navegar por cada uno de ellos y consultar su funcionalidad, descrita de un modo fácil de entender.

Conclusiones

En este capítulo se describió el modelo del sistema del producto. Se describió el diagrama de clases del sistema definiéndose un total de 11 clases del diseño, así como el modelo lógico de datos y el modelo físico de datos. Se explicaron los principios de diseño como los estándares de la interfaz, el tratamiento de errores y la concepción de la ayuda.

Capítulo 4 Estudio de Factibilidad.

Introducción

En el presente capítulo se realiza el estudio de factibilidad del producto utilizando el modelo de Puntos de Función para calcular los costos asociados al proyecto y el tiempo estimado de desarrollo, además se hace un resumen de los beneficios tangibles e intangibles que reportaría el sistema. Por último, con todos los costos y beneficios se analizará si el desarrollo del sistema es factible o no.

Planificación por Puntos de Función

Para realizar el cálculo de los costos de desarrollo del sistema se deben obtener primero las instrucciones fuentes. Analizándose para esto las cantidades de entradas, salidas, peticiones, archivos lógicos e interfaces externas preliminares que tiene el sistema.

Después de un estudio se llegó a los siguientes resultados:

| Nombre de la entrada externa | Cantidad de ficheros | Cantidad de Elementos de datos | de Clasificación(Bajo, Medio y Alto) |
|-------------------------------|----------------------|--------------------------------|--------------------------------------|
| Insertar una familia | 1 | 2 | Bajo |
| Modificar datos de la familia | 1 | 2 | Bajo |
| Eliminar una familia | 1 | 2 | Bajo |
| Insertar una especie | 1 | 8 | Bajo |
| Modificar datos de la especie | 1 | 8 | Bajo |
| Eliminar una especie | 1 | 8 | Bajo |

| | | | |
|------------------------|---|---|------|
| Insertar una ubicación | 1 | 4 | Bajo |
| Eliminar una ubicación | 1 | 4 | Bajo |

Tabla 8: Entradas Externas

| Nombre de la salida externa | Cantidad de ficheros | Cantidad de Elementos de datos | Clasificación(Bajo, Medio y Alto) |
|---|-----------------------------|---------------------------------------|--|
| Mostrar reporte de índices botánicos | 1 | 1 | Bajo |
| Imprimir reporte de índices botánicos | 1 | 1 | Bajo |
| Mostrar reporte de especies con cantidad de ejemplares mayor de x | 1 | 8 | Bajo |
| Mostrar reporte de especies dada la familia | 1 | 8 | Bajo |
| Mostrar reporte de todas las familias con sus especies | 2 | 10 | Medio |

Tabla 9: Salidas Externas

| Nombre de la petición | Cantidad de ficheros | Cantidad de Elementos de datos | Clasificación(Bajo, Medio y Alto) |
|--|-----------------------------|---------------------------------------|--|
| Listar datos de las familias botánicas | 1 | 1 | Bajo |
| Listar datos de las especies botánicas | 1 | 3 | Bajo |
| Listar las ubicaciones de las especies | 1 | 3 | Bajo |
| Imprimir reporte de especies | 1 | 8 | Bajo |

| | | | |
|---|---|----|------|
| con cantidad de ejemplares mayor de x | | | |
| Imprimir reporte de especies dada la familia | 1 | 8 | Bajo |
| Imprimir reporte de todas las familias con sus especies | 2 | 10 | Bajo |
| Buscar familia | 1 | 2 | Bajo |
| Buscar especie | 1 | 8 | Bajo |

Tabla 10: Peticiones

| Nombre del fichero interno | Cantidad de ficheros | Cantidad de Elementos de datos | Clasificación(Bajo, Medio y Alto) |
|----------------------------|----------------------|--------------------------------|-----------------------------------|
| FamiliaBot | 1 | 2 | Bajo |
| EspecieBot | 1 | 8 | Bajo |
| Ubicación | 1 | 4 | Bajo |

Tabla 11: Ficheros Internos

| Elementos | Bajos | X Peso | Medios | X Peso | Altos | X Peso | Subtotal de puntos de función |
|---------------------------|-------|--------|--------|--------|-------|--------|-------------------------------|
| Ficheros lógicos internos | 3 | 7 | 0 | 10 | 0 | 15 | 21 |
| Entradas externas | 8 | 3 | 0 | 4 | 0 | 6 | 24 |
| Salidas externas | 5 | 4 | 1 | 5 | 0 | 7 | 25 |
| Peticiones | 8 | 3 | 0 | 4 | 0 | 16 | 24 |
| Ficheros lógicos | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | |
|----------|--|--|--|--|--|--|----|
| Externos | | | | | | | |
| Total | | | | | | | 94 |

Tabla 12: Puntos de Función

| Características | Valor | |
|--|---------------|--------|
| Puntos de función desajustados | 94 | |
| Lenguaje | SQL | C++ |
| Instrucciones fuentes por puntos de función | 37 | 53 |
| Por ciento de la aplicación en cuanto a requerimientos funcionales | 10% | 90% |
| Instrucciones fuentes | 347,8 | 4483,8 |
| Total de Instrucciones fuentes | 4831,6 | |

Tabla 13: Miles de Instrucciones Fuentes

Costos

Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

| Cálculo de: | Valor | Justificación |
|-------------|-------|---|
| RCPX | 1,00 | BD de tamaño moderada; documentación de acuerdo a las necesidades exactas de las etapas del ciclo de vida; el impacto que tiene una falla en el software es moderado, con perdidas de fácil recuperación; complejidad del software media. |
| RUSE | 1,00 | Se implementa código reusable para el aprovechamiento de este en todo el proyecto. (Nominal) |
| PDIF | 1,00 | No tiene muchas restricciones en cuanto a almacenamiento y recursos; la plataforma de aplicación tiene gran estabilidad.(Nominal) |

| | | |
|------|------|--|
| PERS | 0,63 | No existe movimiento del personal; los programadores y los analistas tienen gran capacidad de comunicación.(Alto) |
| PREX | 0,87 | El equipo posee conocimiento del lenguaje de programación, la plataforma y herramientas utilizadas con una experiencia de aproximadamente un año; tiene experiencia en aplicaciones del tipo de 1 año.(Alto) |
| FCIL | 0,87 | Se utilizan herramientas de programación como: Borland C++ Builder 6, así como la herramienta Rational Rose para la documentación, empleando como notación UML. (Alto) |
| SCED | 1,00 | La planificación se hace con moderada frecuencia. (Nominal) |
| PREC | 3,72 | El equipo de desarrollo posee una comprensión considerable de los objetivos del producto, no tiene experiencia en la realización de software de este tipo. (Nominal) |
| FLEX | 3,04 | El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa. (Nominal) |
| TEAM | 1,10 | El equipo que va a desarrollar el software es altamente cooperativo. |
| RESL | 4,24 | Teniendo en cuenta la alta experiencia que existe en el país acerca de este tipo de estudios existen algunos factores de riesgo. (Nominal) |

| | | |
|------|------|---|
| PMAT | 6,24 | Nivel I Alto porque se encuentra en su primera etapa un poco avanzada. (Bajo) |
|------|------|---|

Tabla 14: Multiplicadores de Esfuerzos y Factores de Escala

Multiplicador de esfuerzos

$$EM = \prod_1^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED$$

$$EM = \prod E_{mi} = 1,00 * 1,00 * 1,00 * 0,63 * 1,00 * 0,87 * 1,00 = 0,548 \approx 0,55$$

Factores de escala

$$SF = \sum SFi = PREC + FLEX + RESL + TEAM + PMAT$$

$$SF = \sum SFi = 3,72 + 3,04 + 4,24 + 1,10 + 6,24 = 18,34$$

Valores de los coeficientes

$$A = 2,94; B = 0,91; C = 3,67; D = 0,24$$

$$E = B + 0,01 * SF$$

$$F = D + 0,2 * (E - B)$$

$$E = 0,91 + 0,01 * 18,34$$

$$F = 0,24 + 0,2 * (1,0934 - 0,91)$$

$$E = 1,0934$$

$$F = 0,27668$$

Esfuerzo

$$PM = A * (MF)^E * EM$$

$$PM = 2,94 * (4,8316)^{1,0934} * 0,55$$

$$PM = 9,05 \text{ (personas meses)}$$

Cálculo del tiempo de desarrollo

$$TDEV = C * PM^F$$

$$TDEV = 3,67 * (9,05)^{0,27668}$$

$$\text{TDEV} = 1,84t$$

Cálculo de la cantidad de hombres

$$\text{CH} = \text{PM} / \text{TDEV}$$

$$\text{CH} = 9.05 / 1,84$$

$$\text{CH} = 4,918 \approx 5$$

Recalculando tiempo de desarrollo para CH=1

Considerando que la tutora y el consultante forman parte del equipo de desarrollo.

$$\text{CH} = \text{PM} / \text{TDEV}$$

$$1 = 9,05 / \text{TDEV}$$

$$\text{TDEV} = 9,05$$

$$\text{TDEV} \approx 9 \text{ meses}$$

Costo

Se asume como salario promedio mensual \$300.

$$\text{CHM} = 1 * \text{Salario Promedio}$$

$$\text{CHM} = 1 * 300$$

$$\text{CHM} = 300 \text{ \$/mes}$$

$$\text{Costo} = \text{CHM} * \text{PM}$$

$$\text{Costo} = \$300 * 9.05$$

$$\text{Costo} = \$ 2715$$

Los costos en los que se incurriría al desarrollarse el sistema serían:

| Cálculo de: | Valor |
|----------------------|-----------------------|
| Esfuerzo(PM) | 9.05 (personas meses) |
| Tiempo de desarrollo | 9 meses |

| | |
|---------------------|--------|
| Cantidad de hombres | 1 |
| Costo | \$2715 |
| Salario medio | \$300 |
| RCPX | 1,00 |
| RUSE | 1,00 |
| PDIF | 1,00 |
| PREX | 1,00 |
| FCIL | 0,87 |
| SCED | 1,00 |

Tabla 15: Resultado de los Costos

Beneficios Tangibles e Intangibles

Los beneficios que brindaría el sistema no son de carácter económico pero si en cuanto a reducción del tiempo de trabajo de los especialistas, orden y control de las especies botánicas. Brinda una notable aceleración en las búsquedas y las tareas que se vayan a realizar. También brindará una mayor comodidad a los especialistas, mejoramiento de las condiciones de trabajo, ahorro en medios y equipos de oficinas y mejoras en la calidad de la información.

Análisis de Costos y Beneficios

El Sistema para el control de Especies y Familias Botánicas del Jardín Botánico Oasis no es un software con fines comerciales, su importancia radica en que resuelve el

problema que hay en el jardín botánico Oasis con el control de las especies y la necesidad de obtener reportes e índices botánicos de manera rápida. El desarrollo del sistema tendría un costo de \$2715 en un tiempo de 9 meses con un desarrollador.

Conclusiones del capítulo

Se realizó el cálculo de factibilidad económica, obteniéndose el esfuerzo, el tiempo y costo que lleva la realización del sistema y se hizo un análisis de los beneficios que aportaría su implantación. Se consideró que por la importancia del sistema para la entidad que utilizará el mismo además que por las características de nuestro sistema socialista de cooperación y solidaridad con todo el mundo y con nuestros compañeros, el equipo de trabajo decidió que era factible el desarrollo del Sistema para el control de Especies y Familias Botánicas del Jardín Botánico Oasis (SCEFBJO).

Conclusiones

El desarrollo del Sistema para el control de Especies y Familias Botánicas del Jardín Botánico Oasis nos ha permitido llegar a las siguientes conclusiones:

- El uso de las tecnologías de la Información y las Comunicaciones es una facilidad que han puesto en las manos del hombre el desarrollo de la humanidad y en un futuro no muy lejano tendrán un papel aún más protagónico en la sociedad.
- Se hizo un estudio del estado del arte comprobando que en el país no había una aplicación que satisficiera con todos los requerimientos de los especialistas.
- Se decidió desarrollar el sistema en el entorno de desarrollo rápido de aplicaciones Borland C++ Builder versión 6 utilizando el lenguaje de programación C++ y como gestor de bases de datos se escogió MS Office Access 2003 después de haber hecho un estudio de las principales tecnologías existentes hoy día.
- Se desarrolló un sistema que permitió el trabajo con las especies y familias botánicas, así como la importación y exportación de la base de datos a otros soportes informáticos.

Recomendaciones

El objetivo de este trabajo es crear un sistema informático que facilite el trabajo con las especies y familias botánicas, por eso, como recomendaciones se propone la adición de técnicas de inteligencia artificial para decidir en que parte del jardín es recomendable ubicar la especie. Para esto se necesitarían datos y estudios que no están contemplados en este trabajo, como por ejemplo la acidez del suelo, la humedad, la cantidad de luz, la competencia de una especie con otra, etc. Con estos estudios y teniendo en cuenta las características propias de cada especie el sistema podría inducir en que lugar sembrar la planta en caso de una nueva adquisición del jardín.

Referencias Bibliográficas

- [1] J. Rosario, "La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual," 2005.
- [2] P. M. Graells, "LAS TIC Y SUS APORTACIONES A LA SOCIEDAD. ," 2008.
- [3] Microsoft Corporation, "Botánica," in *Microsoft® Student 2008 [DVD]*.
Microsoft ® Encarta ® 2008. © 1993-2007 Microsoft Corporation. Reservados todos los derechos., 2007.
- [4] Microsoft Corporation, "Jardín botánico," in *Microsoft® Student 2008 [DVD]*.
Microsoft ® Encarta ® 2008. © 1993-2007 Microsoft Corporation. Reservados todos los derechos., 2007.
- [5] Rational Software, Microsoft, Hewlett-Packard, Oracle, Sterling Software, MCI Systemhouse, Unisys, ICON Computing, IntelliCorp, i-Logix, IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies, and Softeam, "UML Summary," 1997.
- [6] P. Kruchten, "The Rational Unified Process An Introduction, Second Edition," A. Wesley, Ed.: Addison Wesley, 2000.
- [7] D. Antón, M. G. Pérez, and M. Angélica, "Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática," 2004.
- [8] H. M. Deitel - Deitel & Associates Inc. and P. J. Deitel - Deitel & Associates Inc., "C++ How to Program, Fifth Edition," Prentice Hall, 2005.
- [9] Microsoft Corporation, "Java (Informática)," in *Microsoft® Student 2008 [DVD]*.
Microsoft ® Encarta ® 2008. © 1993-2007 Microsoft Corporation. Reservados todos los derechos., 2007.
- [10] W. Huamani, "Introducción a Java.," L. n. D. R. 2007, Ed., 2007.
- [11] C. Walnum, "Java by example.," Q. Corporation, Ed., 1996.
- [12] C. Casares, "Tutorial de SQL.," 2004.
- [13] D. P. Bello and A. M. Capote, "Sistema de Evaluación de Cualidades Dinámicas y de Consumo," in *Facultad de Informática Cienfuegos: Universidad de Cienfuegos*, 2006.

- [14] "MySQL toma nuevo impulso y ya es una seria amenaza para Oracle, IBM y Microsoft.," 2005.
- [15] Microsoft Corporation, "Especie," in *Biblioteca de Consulta Microsoft® Encarta®*, 2005.
- [16] Microsoft Corporation, "Familia," in *Biblioteca de Consulta Microsoft® Encarta®*, 2005.
- [17] Rational Software Corporation, "Rational Unified Process Best Practices for Software Development Teams," 1998.
- [18] I. Jacobson, G. Booch, and J. Runbaugh, "The Unified Modeling Language Reference Manual," Addison Wesley, 1999.
- [19] S. S. Alhir, "Learning UML," O'Reilly, 2003.
- [20] "Comunidad de desarrolladores. Diagrama de Clases de Diseño.," 2006.

Bibliografía

- J. Cogswell, C. Diggins, R. Stephens, and J. Turkanis, "C++ Cookbook," O'Reilly, 2005.
- "El motor de Base de Datos para la Empresa," 2007.
- I. Jacobson, G. Booch, and J. Runbaugh, El Proceso Unificado de Desarrollo de Software, 2000.
- E. N. Barroso and M. R. Andino, "Las TIC y su papel en los procesos de Innovación en el ámbito económico local.," 2000.
- S. S. Alhir, "Learning UML," O'Reilly, 2003.
- D. Axmark, M. Widenius, and K. Aldale, "MySQL Reference Manual," 1998.
- Borland Software Corporation, "Programing with C++ Builder," 2002.
- C. Larman, UML y Patrones: Prentice Hall Hispanoamericana, S.A., 1999.
- Borland Software Corporation, "What's new in C++ Builder," 2002

Anexos

Prototipos

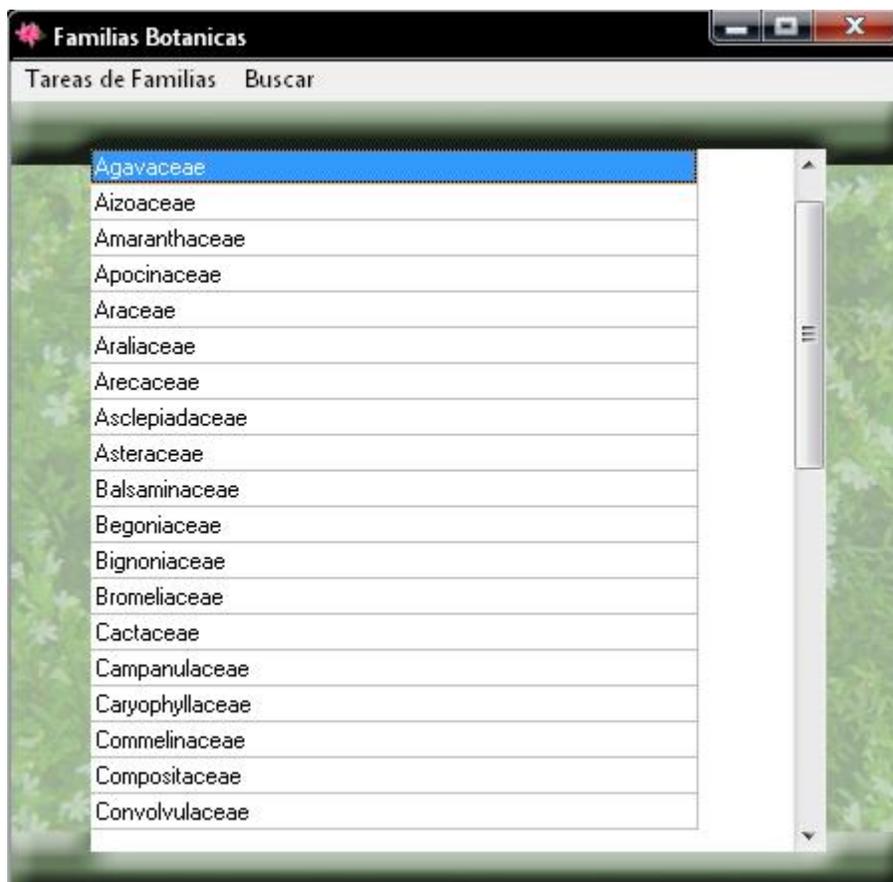


Figura 7 Prototipo Familias

Agregar Familia Botanica

Nombre de la Familia Botanica:

Descripciones de la Familia:

Agregar

Figura 8 Prototipo Insertar Familia

Modificar datos de Familia Botanica

Nuevo Nombre de Familia

Nuevas Caracteristicas

La familia Apoc|

Aceptar

Figura 9 Prototipo Modificar Familia



Figura 10 Prototipo Ver Familia

Especies Botánicas

Tareas de Especies Buscar Mapa

| Nombre Científico | Nombre Vernáculo | Familia Botanica |
|---------------------------------------|----------------------------|------------------|
| Agave marginata | Agave | Agavaceae |
| Yucca aloifolia L. | Balloneta | Agavaceae |
| Yucca elephantipes | yuca | Agavaceae |
| Beloperone guttata | Camarón rojo | Acanthaceae |
| Crossandra infundibuliformis (Brasil) | Cosandra | Acanthaceae |
| Fittonia verschaffeltii | Fitonia roja | Acanthaceae |
| Fittonia verschaffeltii argyroneura | Fitonia | Acanthaceae |
| Graptophyllum pictum | Eranthemus | Acanthaceae |
| Graptophyllum pictum 'Tricolor' | Eranthemus rojo | Acanthaceae |
| Hemigraphis colorata | Mosaico | Acanthaceae |
| Hypoestes sanguinolenta (rosada) | Mosaico 1 | Acanthaceae |
| Justicia ovata | Justicia | Acanthaceae |
| Megakepasma erythrochlamys | Falsa flor del Brasil | Acanthaceae |
| Pachystachys lutea | Camarón amarillo | Acanthaceae |
| Pseuderanthemum atropurpureum | Eranthemus (rojo completo) | Acanthaceae |
| Ruellia colorata | Camarón rojo 1 | Acanthaceae |
| Thunbergia erecta | Maena, Meyinea | Acanthaceae |
| Thunbergia grandiflora 'Alba' | Fausto enredadera blanco | Acanthaceae |

Figura 11 Prototipo Especies

Agregar Especie

Nombre Científico de la Especie

Nombre Vernaculo de la Especie

Familia a la que pertenece

Cantidad de ejemplares

Origen de la Especie

Características propias de la Especie

NO HAY IMAGEN CARGADA

Agregar Especie

Figura 12 Prototipo Insertar Especie

Modificar datos de una Especie

Nombre Científico de la Especie

Nombre Vernaculo de la Especie

Familia a la que pertenece

Cantidad de ejemplares

Origen de la Especie

Dia de Entrada (dd/mm/aaaa)

Características propias de la Especie

Modificando esp



Guardar Cambios

Figura 13 Prototipo Modificar Especie

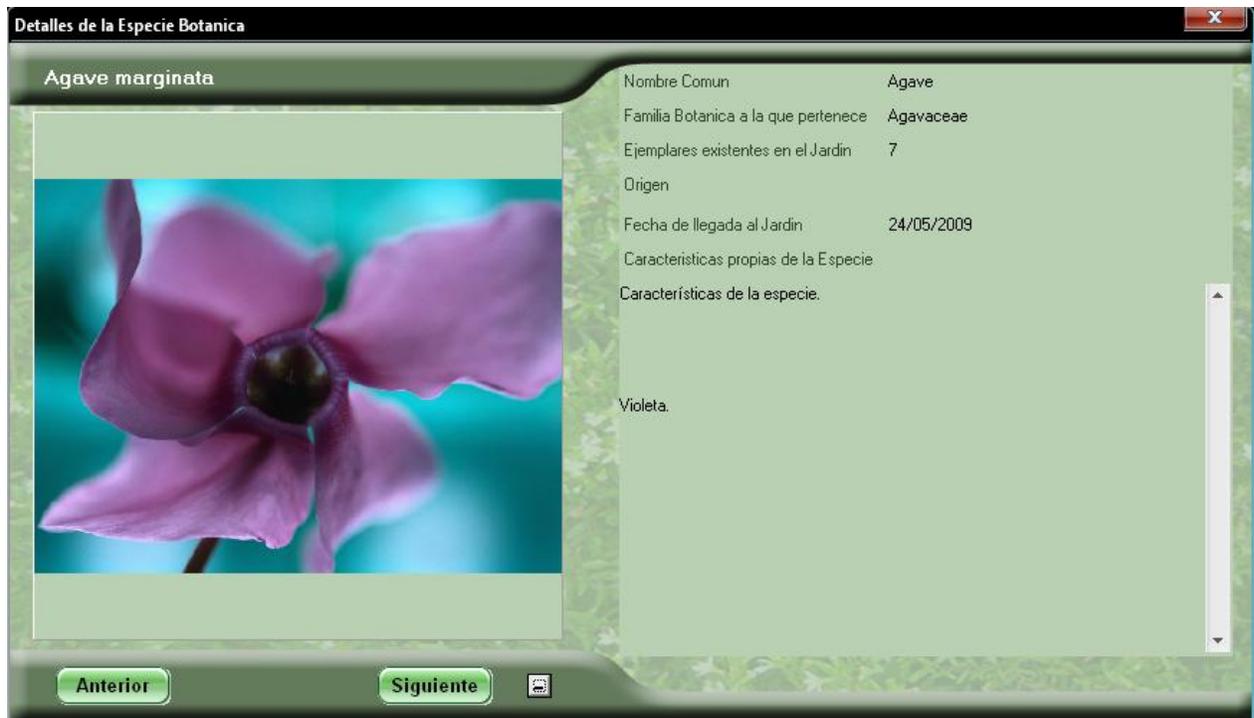


Figura 14 Prototipo Ver Especie

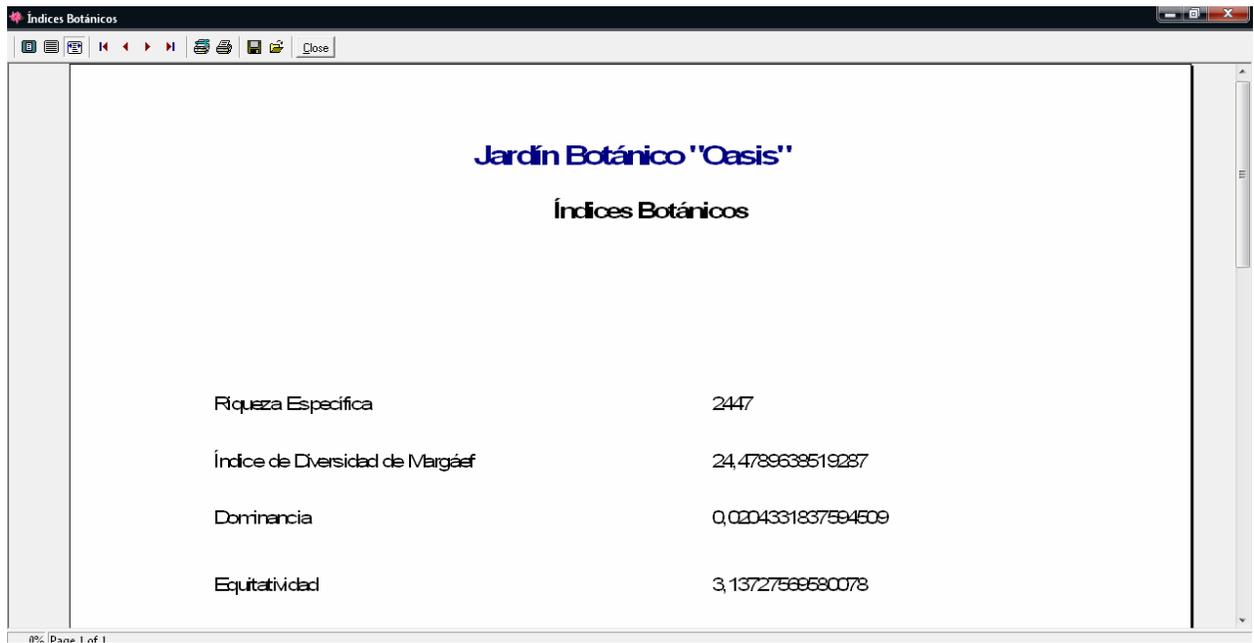


Figura 15 Prototipo Reportes



Figura 16 Prototipo Mostrar Mapa