

PROTOTIPO, EL CONTEXTO Y LA INGENIERIA DEL SOFTWARE

Emma Corina Salazar Suárez, ecorina@yahoo.com
Estudios de Postgrado, en Sistemas de Información
Universidad Católica Andrés Bello, Caracas, Venezuela.

Resumen:

El concepto de prototipo tiene varios significados en el contexto, desde su uso en el lenguaje común, mediante la comunicación diaria o cotidiana pasando por las precisiones que ha hecho la lingüística, y que amplían su interpretación, siendo que se comenzó a usar con mayor énfasis, a partir de los años 50. Según su uso, y específicamente dentro del área de ingeniería del desarrollo de sistemas, diseño, creación de programas y aplicaciones, la construcción de prototipos es un procedimiento posible desde el desarrollador; aunque autores como Carr y Verner, lo entienden como un paso inevitable y necesario en relación a considerar los beneficios que se han establecido producen, como entender bien los requerimientos, considerando la existencia de la metodología de desarrollo de prototipo, con matices en su utilización, que van desde tablas borrador, el desarrollo rápido, el desarrollo incremental al evolutivo. Otros autores, como Floyd, resaltan la manera de dar cabida a las necesidades del usuario y de la directiva, hasta llegar a ser definitiva la razón de ser, en el desarrollo por la ingeniería de la usabilidad. El prototipo es algo tangible, funcional y aunque también se indica que se tiene que concentrar en lo que debe hacer el sistema y no en como lo hace, lo cual a veces involucra una contradicción, tiene que incluir al menos, una notación gráfica, información y posible comportamiento del sistema. Como se muestra, hay variados tipos de prototipos, siendo que construir un prototipo, tiene sus ventajas y desventajas. Se utiliza contenido de fuentes académicas, y no convencionales halladas en la web, sobre su utilización.

Palabras claves: prototipo, contexto, ingeniería del software, usabilidad, sistemas de información.

1 INTRODUCCION

Hay distintos conceptos de lo que se denomina y espera de un prototipo, en la medida estamos hablando en lenguaje común, desde la etimología, la lingüística, la ingeniería física o constructiva, y específicamente de la ingeniería del software, en la eficaz generación de sistemas de información, para operaciones, clasificar conocimientos, reportar estadísticas, hacer escenarios, evaluar la gestión; en la actualidad, posiblemente hasta llegar a vincular la data con su proximidad o pertenencia geográfica, temporal o espacial, e incluso contemplar, la vinculada a los controles biométricos, cambio meteorológicos o nanomoleculares.

El extenso uso de la palabra *prototipo*, pareciera tener un origen sintetizado a efectos comunicacionales, como señalando los antecedentes de inventos, innovaciones, como parte de la labor del ingenio humano, a veces de fecha poco exacta¹, por ende, pudiéramos pensar, desde un hipotético dibujo antes de hacer la rueda, a los diseños de la maquina voladora de la

Leonardo Da Vinci, precedente del helicóptero, pero a pesar de su uso frecuente, como vocablo, es impreciso su origen, más allá de lo que se señala que proviene de dos vocablos de raíces griegas, *proto* relativo al “primero” y *tipo*, referido a “impresión, modelo”².

Sin embargo, adentrándonos en el contexto donde se comenzó a distinguir lo que es la ingeniería del software y a usar el vocablo, podemos entender que lo que pareciera referirse a hacer o realizar un patrón único, para algunos “modelo”, esto dista de ser algo unívoco, ya que su uso comporta reflexiones multidisciplinarias, y de mucho ingenio, horas de labor individual o de equipos, que han tratado de ser estandarizadas, a través de diversos métodos y bajo una metodología, que es enunciada en distintas formas, y que finalmente, se incorporan a las herramientas de la ingeniería del software, con parecidos usos, en otros campos de conocimiento.

A continuación, se reflexiona sobre que es un prototipo, la metodología orientada a la construcción de prototipos, sus características ideales, algunas herramientas de prototipado, así como las ventajas y desventajas que se señalan en su uso, en la práctica del actual desarrollo de sistemas, revisando literatura especializada, y sitios de consulta muy frecuentes, a través de la navegación en internet.

2 DE LOS PROTOTIPOS

En la literatura ingenieril, el uso del vocablo prototipo, está primeramente vinculado a la construcción de los zepelines, a principios de siglo³, y posteriormente, de las expresiones sobre los procedimientos para la invención de las computadoras⁴, siendo que en su uso actual, tiene un doble significado, como lo notaremos a continuación.

En consulta al *Diccionario de la Real Academia*, encontramos en principio, 2 significados, el primero vinculado a realizar algo tangible, como: **1.** “Ejemplar original o primer molde en que se fabrica una figura u otra cosa”; y el segundo, en relación dialéctica, como complemento, contradicción o supuesto distinto, como a una condición: **2.** “Ejemplar más perfecto y modelo de una virtud, vicio o cualidad”⁵.

Este doble énfasis, en aspectos físicos o en la calificación de su conformación, es de uso común, ya que conceptos semejantes son establecidos por el *Diccionario Manual de la Lengua Española*, con los añadidos de ejemplos, como:

“**1** Primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otros iguales, o molde original con el que se fabrica: *un prototipo de coche eléctrico*. **2** Persona o cosa que reúne en grado máximo las características principales de cierto tipo de cosas y puede representarlas: *esa chica es el prototipo de belleza*”⁶.

Por otra parte, desde el punto de vista de la Ingeniería del Software, se exalta su consideración como: Un prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas⁷.

También debemos considerar, lo que hacen ver los autores Carr y Verner, los prototipos se utilizan en muchas disciplinas, y en particular por los Ingenieros⁸, en las distintas ramas de la Ingeniería, en esa función de primer tipo:

“...por ejemplo, los ingenieros de fabricación de los productos hacen prototipos para explorar y controlar la incertidumbre en el diseño de un producto, o con el fin de explorar las dificultades en el proceso de producción antes de la producción o fabricación a gran escala fabricación del producto. La industria del software ha adoptado esta técnica industrial para la construcción de prototipos como modelos, simulaciones, o como para implementaciones parciales de sistemas y a los utilizan para una variedad de propósitos diferentes, por ejemplo, para probar la viabilidad de algunos aspectos técnicos de un sistema, o como especificación herramientas para determinar los requisitos del usuario”⁹.

Otros innovadores, como los diseñadores, señalan que estudiar los requerimientos puede llevar a una propuesta diferente que va más acorde con lo que en principio expresa un usuario, y que ese es el cometido de su labor, al obtener o procurar una satisfacción mayor, siendo que para ellos, el prototipo es:

“... primer modelo del producto a tamaño real y con los materiales definitivos planteados para su fabricación, que tiene como propósito la validación técnico – constructiva, funcional, ergonómica, de distribución y/o mercado de la propuesta”¹⁰.

Finalmente tenemos aquellos, que integran varias perspectivas y lo igualan a: “Modelo o maqueta del sistema que se construye para comprender mejor el problema y sus posibles soluciones”¹¹. Esto se considera permite:

- Evaluar mejor los requisitos.
- Probar opciones de diseño.

3 EL CONTEXTO

El uso de los prototipos, sin embargo, no es un uso reservado al área de la ingeniería, sino como se evidencia también se usan dentro de la estética, de la salud, de la comunicación, los estudios de mercado y sociales, justo cuando se elaboran los instrumentos e inclusive grupos “pilotos”, al adentrarse en el mundo de las representaciones sociales, y por ende, de como ha sido entendido y desarrollado su uso para la ingeniería del software, en su relación dentro de la misma disciplina, así como desde otras ciencias y prácticas.

En este sentido, aunque poco señalado, mucho del uso y formas de entender los prototipos, es desde la filosofía del lenguaje, la lingüística y la teoría de la comunicación, en posiciones establecidas a mediados de los años 60, paralelas como veremos al origen de la Ingeniería de Software, y a veces poco relacionadas, en el auge en los estudios de construcciones de maquetas, herramientas visuales, y posteriormente de los mapas, en particular de los mapas mentales, y otras representaciones alternas a los diagramas, flujogramas y anteriores a las metodologías orientadas a objetos.

De este modo su uso sería paralelo al desarrollo de corrientes de pensamiento, tan importantes como la que lleva a la enunciación y comprensión de la teoría de los actos del habla,

con Austin, en cuanto al locucionario, el ilocucionario y el perlocucionario¹², mientras que otros lo establecen desde la psicología o la lingüística cognitiva, con los trabajos de la psicóloga Eleonor Rosch, y posteriores desarrollos del tema, superación de la comprensión del cerebro solo como bilobular y comprensión del cerebro triuno, que van generando explicaciones diversas, en cuanto a la pertinencia de las construcciones de las categorizaciones, en el pensamiento, donde se llega a la medida que una sola opción o dimensión, pudiera tener más de un significado, y que "medir" lo que se dice, lo dicho comporta al menos 3 elementos en simultanea acción, en relación a lo comunicado, lo que se emite y lo que entiende el receptor.

Todo esto abre una interesante perspectiva, sobre la cual solo se exponen estos breves elementos a consideración, pero que permiten entender que por ello, ha sido considerado cada vez más importante la elaboración de prototipos, en cuanto a lo que abarca como patrón como palabra y en su uso diario, siendo que incluso la construcción de un solo prototipo comporta más de un significado, que puede ser diferenciados y caracterizados.

Por ende, vemos como todo prototipo, puede ser entendido desde: 1) lo que pretende hacer; 2) lo que se hizo para hacerlo; 3) el resultado de lo realizado y 4) lo que se considera se puede hacer de manera diferente, aspectos que van a entrar en conjunto a ser evaluados, y que dan pie a los distintos aspectos que han sido resaltados en estas tres últimas décadas, desde la popularización del término.

4 LA INGENIERIA DEL SOFTWARE

La **Ingeniería del Software** es la "rama de la ingeniería que crea y mantiene las aplicaciones de software usando tecnologías y prácticas de las ciencias de la computación, manejo de proyectos, el ámbito de la aplicación, y otros campos"¹³. Disciplina que trata los aspectos concernientes al desarrollo de sistemas complejos y de calidad, que requiere para su construcción de: un equipo de personas, procesos rigurosos, sistemáticos y controlables (modelos y métodos) y la utilización de herramientas¹⁴.

El término *Ingeniería*, se define en el Diccionario de la Real Academia Española de la Lengua como: "1. f. Estudio y aplicación, por especialistas, de las diversas ramas de la tecnología; 2. f. Actividad profesional del ingeniero", por ende, en un primer lugar da cabida al ejercicio de quien pueda tener el suficiente ingenio, conocimiento para manejar máquinas, herramientas y generar conocimiento, y solucionar problemas, crear artefactos.

Según la definición del Instituto de Ingenieros Eléctricos y Electrónicos, IEEE, que reúne a expertos del área y otras profesiones afines, el "*software es la suma total de los programas de ordenador, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo*" y "*un producto de software es un producto diseñado para un usuario*". En este contexto, la Ingeniería de Software (SE del inglés "*Software Engineering*") es un "enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software"¹⁵.

Por su parte, se señala fue Fritz Bauer, quien en la primera conferencia sobre desarrollo de software patrocinada por el Comité de Ciencia de la OTAN celebrada en Garmisch (Alemania), en octubre de 1968, quien utiliza el término *Ingeniería del Software*; previamente también se sostiene que el holandés Esdger Dijkstra lo usa, en su obra *The Humble Programmer*. En este sentido, según Alan Davis, se usó como: "*la aplicación inteligente de principios probados, técnicas, lenguajes y herramientas para la creación y mantenimiento, dentro de un coste razonable, de software que satisfaga las necesidades de los usuarios*"¹⁶.

Además, se señalaba que la ingeniería del software, surge a mediados de estos años, en medio de la carrera por el control del espacio y la necesidad de desarrollos de sistemas a tiempo, porque el contexto de desarrollo, adolecía de:

- Retrasos considerables en la planificación
- Poca productividad
- Elevadas cargas de mantenimiento
- Demandas cada vez más desfasadas frente a las ofertas
- Baja calidad y fiabilidad del producto
- Dependencia de los realizadores

5 CONSIDERACIONES PRELIMINARES AL DESARROLLO DEL PROTOTIPO

Antes de decidir, si se va a desarrollar un prototipo, en cuanto si debe incluirse o no en un Ciclo de Desarrollo de Sistema de Información, del trabajo inédito de Pereyra¹⁷, que recoge unas consideraciones que el profesional a cargo, tiene que evaluar para la realización de un prototipo, como en los casos de:

- Problemas no estructurado, novedosos y complejos, de información personalizada del usuario, ya que sus salidas no son predecibles y definidas
- Problemas de ambiente Inestable, el profesional también debe evaluar el contexto del sistema
- Experiencia en diseños similares
- No se conocen los requerimientos, la naturaleza del sistema es tal que existe poca información con respecto a las características que debe tener el nuevo sistema para satisfacer las necesidades del usuario
- Los requerimientos deben evaluarse, se conocen los requerimientos aparentes de información pero es necesario verificarlos y evaluarlos
- Costos altos, donde la inversión involucra gran cantidad de recursos financieros y humanos.
- Altos riesgo, la evaluación inexacta de los requerimientos o el desarrollo incorrecto ponen en peligro a la organización
- El usuario no está dispuesta examinar modelos en papel, o no sabe lo que quiere pero lo reconocerá cuando lo vea.
- Tecnologías Nuevas: la falta de experiencia en el uso de dichas tecnologías, junto con el deseo de instalar nuevas tecnología hace que sea propicio el uso del prototipo.

6 METODOLOGIAS DE DESARROLLO SOFTWARE POR PROTOTIPOS

Las Metodologías de Desarrollos de Sistemas por Prototipos, “nos señalan en realidad varios métodos para que funjan como herramientas útiles para agilizar el proceso de obtención de una solución de sistemas requerida por cualquier organización, en especial, si la organización no cuenta con mucho tiempo”¹⁸.

Estas metodologías, y aquellas que reivindican la construcción de prototipos, se crea como una revisión a varios modelos de desarrollo, empezando por el modelo de cascada clásico, y luego dentro del ciclo de vida de desarrollo de software; también se identifican con la labor de desarrollo rápido, del modelo iterativo e incremental.

Sin embargo, a veces también se habla de todas estas metodologías, como un solo modelo, el cual debe ser cuidadosamente llevado, en el que deben establecer límites reales a lo que se quiere desarrollar, previendo cuantas veces el producto final tendrá que pasar repetidamente por el proceso de desarrollo, ajustes y revisión, mientras se logra adaptar el modelo fielmente a los requerimientos iniciales o se desarrollen las funcionalidades solicitadas. El desarrollo se relaciona con el enfoque que se escoge para abordar la construcción del prototipo.

7 ENFOQUES PARA LA REALIZACIÓN DE UN PROTOTIPO

Volviendo a los autores Carr y Verner, el proceso de construir un prototipo, puede representarse generalmente, como 3 pasos, o más detalladamente, como : 1) la elaboración de un patrón, modelo o propuesta; 2) que es revisado por los usuarios; 3) se decide si satisface gran parte de las condiciones o todos los requerimientos; 4) si hay observaciones o no satisfacción, 5) se revisa o refina el prototipo, 6) si se decide que si cumple con el cometido, 7) se consideran, los objetivos alcanzados.

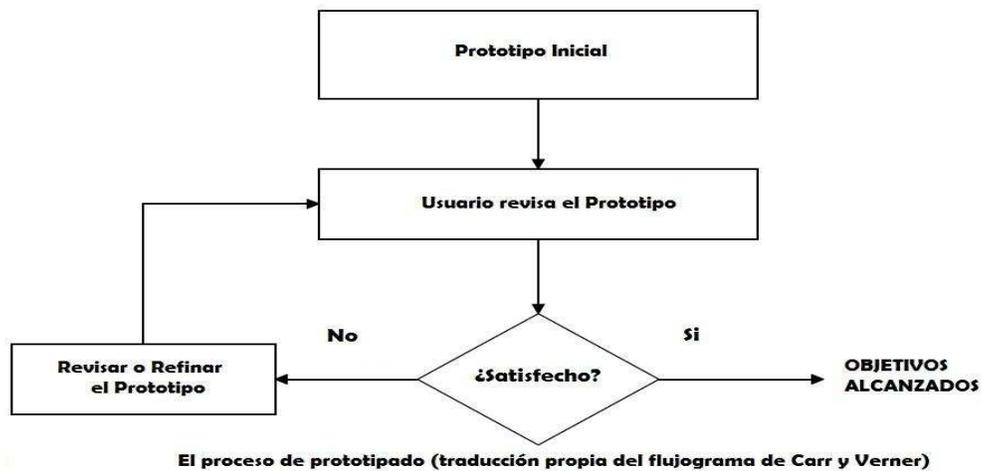


Fig. 1 El proceso de Prototipado

Según los autores señalados, Carr y Verner, basados en el trabajo realizado por Floyd¹⁹, a principios de los 90, la construcción del prototipo va muy ligado al tipo de “producto” que se desea realizar o entregar, de acuerdo, al nivel de desarrollo planteado, al área, al tipo de investigación, donde se distinguen distintos momentos en que puede ser elaborado y presentado.

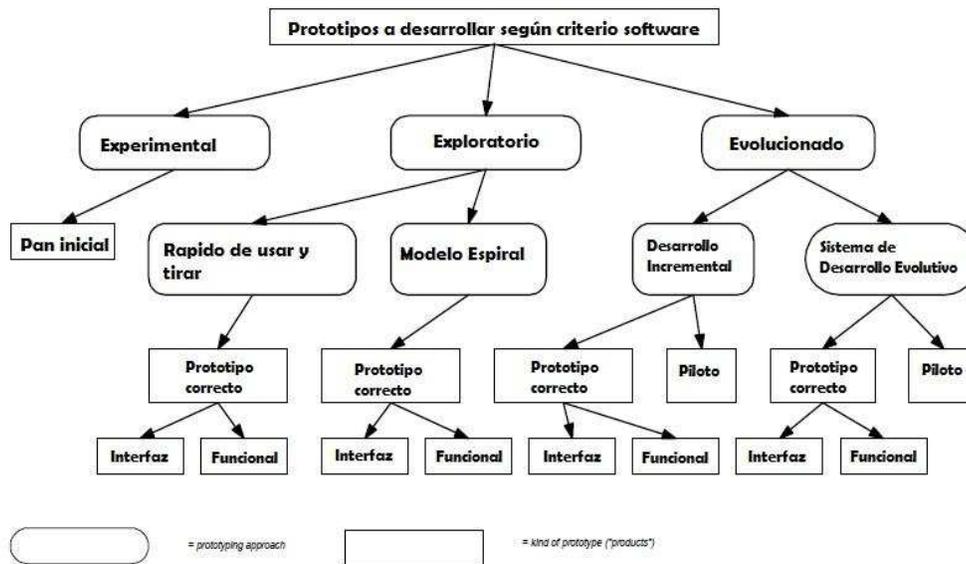
En este sentido, se clasifica que los enfoques de desarrollo de software, emplean prototipos que pueden estar basados en uno de los siguientes objetivos:

- 1) De exploración,
- 2) Experimentación y
- 3) Evolución.

Los enfoques de tipo *exploratorio* son aquellos que se realizan principalmente para obtener o clarificar usuario requisitos. Ayudan a los desarrolladores a obtener una perspectiva de los usuarios tareas y problemas, y a cristalizar las percepciones confusas del usuario y de las necesidades en los requisitos para un sistema inicial.

Cuando hablamos de prototipos *experimentales*, es cuando se proporciona a los usuarios distintas posibilidades y alternativas; ayuda a evaluar la viabilidad del futuro sistema de términos de rendimiento y otros aspectos técnicos, cuando los requisitos son conocidos.

Por otra parte, el prototipado *evolutivo* es un enfoque que permite flexibilidad en el software el desarrollo de procesos para que pueda adaptarse a contextos cambiantes de la organización. Es decir, el proceso de desarrollo de software no es visto como un hecho aislado autónomo de proyecto, sino como algo que evoluciona continuamente. Se representan de la siguiente manera:



Prototipos, según enfoques de desarrollo software / traducción propia de diagrama de Carr y Verner 1995 / Propuesta Floyd - 1984

Fig. 2 Enfoques de prototipos, según criterio software.

Trabajos más recientes, de la década del 2000, siglo XXI, añaden a esta clasificación, como enfoques para la construcción de prototipos, las categorías de:

- Vertical: Desarrolla completamente alguna de las funciones.
- Horizontal: Desarrolla parcialmente todas las funciones.
- Iterativo: se repite el proceso hasta completar nivel²⁰
- Orientado a la Tarea: En relación directa a lo que se desea hacer
- Basado en Escenario: En relación a evaluar lo que hace
- Fijo: Las categorías involucradas son establecidas y poco variables
- Abierto: Las categorías, se pueden variar, cambiar, eliminar

Otros autores, como Rivas, Poletti, y Salinas, siguiendo a Pressman, clasifican la elaboración de prototipos, como productos de 2 enfoques, el cerrado y el abierto, siendo cada uno de ellos:

- **“El enfoque cerrado o prototipo desechable.** Que sirve básicamente para capturar, analizar y validar requerimientos del sistema. Posee un período de vida corto. Luego se desecha y se realiza una ingeniería del software con un paradigma diferente de acuerdo a las especificaciones obtenidas con el prototipo.
- **El enfoque abierto o prototipo evolutivo.** El cual emplea el prototipo como primera parte de una actividad de análisis a la que seguirá el diseño y la construcción del sistema. El prototipo del software es la primera evolución del sistema terminado”²¹.

8 TIPOS DE PROTOTIPO

Los prototipos pueden tomar muchas formas: Productos, metáforas, guiones gráficos, pantallas pintadas, pantallas animados y programas; en Alta o Baja tecnología; fidelidad.

- Son prototipos de alta tecnología, aquellos que se centran en la presentación. Si bien son útiles en este contexto, que tienden a centrarse en los desarrolladores de estas cuestiones demasiado temprano en el proceso de diseño. También tienden a fomentar una imagen del prototipo, como subproducto, y llevar a la suposición implícita de que el diseño ya está completo. En su lugar, creación de prototipos debe ser una exploración conjunta de funciones entre los desarrolladores y los usuarios.
- De baja tecnología: ofrece una alternativa a las herramientas de creación de prototipos complejos y costosos. Prototipos de baja tecnología se crean con papel, tijeras, bolígrafos o lápices de colores, y la animación manual. Usted puede falsificar casi cualquier técnica de interfaz de usuario (con el Mago de Oz técnica: ". No prestes atención al hombre detrás de la pantalla"), y puede incluir el mundo exterior del prototipo. Esto hace que sea fácil de probar múltiples ideas muy rápidamente. Además, dejando detalles de estos primeros prototipos obliga a los evaluadores a centrarse en el panorama general.

Para Meza y Barcasnegras, los prototipos según tipos pueden ser: 1) primer tipo, 2) corregidos; 3) funcionales, y 4) de características seleccionadas²².

Otras variantes de prototipo, es según su uso inmediato o por sus funciones, como:

- Prototipos por Pantallas
- Prototipos para Procedimientos de Procesamientos
- Prototipos de Funciones Básicas²³.

8 CARACTERISTICAS IDEALES DE LOS PROTOTIPOS

De las lecturas realizadas, podemos decir como características ideales de los prototipos:

- ✓ El prototipo es una aplicación que funciona
- ✓ Los prototipos se crean con rapidez
- ✓ Los prototipos evolucionan a través de un proceso iterativo
- ✓ Los prototipos tienen un costo bajo de desarrollo
- ✓ Permiten que se cree correctamente el sistema deseado

9 PASOS O FASES PARA LA CONSTRUCCION DE UN PROTOTIPO :

El desarrollo de un prototipo se lleva a cabo en forma ordenada a través de diversas etapas, que van en relación a la metodología de software utilizado, y el autor tomado como guía, siendo enunciadas, las siguientes etapas:



Fig. 3 El Modelo de desarrollo Orientado a Prototipos
(Tomado de <http://sistemas2009unl.wordpress.com/prototipos-informaticos/>)

Las fases que comprende el método de desarrollo orientado a prototipos serían:

- **Investigación preliminar.** Las metas principales de esta fase son: determinar el problema y su ámbito, la importancia y sus efectos potenciales sobre la organización por una parte y, por otro lado, identificar una idea general de la solución para realizar un estudio de factibilidad que determine la factibilidad de una solución software.
- **Definición de los requerimientos del sistema.** El objetivo de esta etapa es registrar todos los requerimientos y deseos que los usuarios tienen en relación al proyecto bajo desarrollo. Esta etapa es la más importante de todo el ciclo de vida, es aquí donde el desarrollador determina los requisitos mediante la construcción, demostración y retroalimentaciones del prototipo. Por lo mismo esta etapa será revisada con más detalle luego de esta descripción.
- **Diseño técnico.** Durante la construcción del prototipo, el desarrollador ha obviado el diseño detallado. El sistema debe ser entonces rediseñado y documentado según los estándares de la organización y para ayudar a las mantenciones futuras. Esta fase de diseño técnico tiene dos etapas: por un lado, la producción de una documentación de diseño que especifica y describe la estructura del software, el control de flujo, las interfaces de usuario y las funciones y, como segunda etapa, la

producción de todo lo requerido para promover cualquier mantención futura del software.

- **Programación y prueba.** Es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos.
- **Operación y mantención.** La instalación del sistema en ambiente de explotación, en este caso, resulta de menor complejidad, ya que se supone que los usuarios han trabajado con el sistema al hacer las pruebas de prototipos. Además, la mantención también debería ser una fase menos importante, ya que se supone que el refinamiento del prototipo permitiría una mejor claridad en los requerimientos, por lo cual las mantenciones perfectivas se reducirían. Si eventualmente se requiriese una mantención entonces el proceso de prototipado es repetido y se definirá un nuevo conjunto de requerimientos.

La fase más importante corresponde a la definición de requerimientos, la cual correspondería a un proceso que busca aproximar las visiones del usuario y del desarrollador mediante sucesivas iteraciones. La definición de requerimientos consiste de cinco etapas entre dos de las cuales se establece un ciclo iterativo:

- **Análisis grueso y especificación.** El propósito de esta subfase es desarrollar un diseño básico para el prototipo inicial.
- **Diseño y construcción.** El objetivo de esta subfase es obtener un prototipo inicial. El desarrollador debe concentrarse en construir un sistema con la máxima funcionalidad, poniendo énfasis en la interface del usuario.
- **Evaluación.** Esta etapa tiene dos propósitos: extraer a los usuarios la especificación de los requerimientos adicionales del sistema y verificar que el prototipo desarrollado lo haya sido en concordancia con la definición de requerimientos del sistema. Si los usuarios identifican fallas en el prototipo, entonces el desarrollador simplemente corrige el prototipo antes de la siguiente evaluación. El prototipo es repetidamente modificado y evaluado hasta que todos los requerimientos del sistema han sido satisfechos. El proceso de evaluación puede ser dividido en cuatro pasos separados: preparación, demostración, uso del prototipo y discusión de comentarios. En esta fase se decide si el prototipo es aceptado o modificado.
- **Modificación.** Esto ocurre cuando la definición de requerimientos del sistema es alterada en la sub-fase de evaluación. El desarrollador entonces debe modificar el prototipo de acuerdo a los comentarios hechos por los usuarios.
- **Término.** Una vez que se ha desarrollado un prototipo estable y completo, es necesario ponerse de acuerdo en relación a aspectos de calidad y de representación del sistema.

Desarrollo de un Modelo

En esta etapa se explica el método iterativo y las responsabilidades a los usuarios ya que el usuario participa directamente en todo el proceso. La rapidez con la que se genera el sistema es esencial para que no se pierda el estado de ánimo sobre el proyecto y los usuarios puedan comenzar a evaluar la aplicación con la mayor brevedad posible. El

profesional de sistema para construcción inicial del prototipo emplea cualquier herramienta, como Lenguajes de Cuarta Generación, Generadores de Reportes, Generadores de Pantallas. En el desarrollo de un prototipo se preparan los siguientes componentes:

- El lenguaje para el diálogo o conversación entre el usuario y el sistema
- Pantallas y formatos para la entrada de datos
- Módulos esenciales de procesamientos
- Salida del sistema

La incorporación en la interfaz de entrada/salida de características representativas de las que serán incluidas en el sistema final permite una mayor exactitud en el proceso de evaluación.

Revisión del Prototipo

Es responsabilidad del usuario trabajar con el prototipo y evaluar sus características y operación. La experiencia con el sistema bajo condiciones reales permite la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios, o también la eliminación de características innecesarias.

Iteración

Los dos últimos etapas descritas anteriormente se repiten varias veces hasta que estén usuarios y profesionales de sistema de acuerdo en que el prototipo ha evolucionado lo suficiente o que una iteración más no traerá beneficios adicionales.

Prototipo Terminado

Cuando el prototipo está terminado, es decir, tenemos la información que buscamos seguimos en el punto donde habíamos quedado dentro del Ciclo de Desarrollo de Sistema.

Por otra parte, Sonia Marinó, basada en Mariño y Godoy (2008) atendiendo a especificaciones para el desarrollo de prototipos incrementales o evolutivos en el ámbito educativo (Corcos, 2000 y Oliveros, 2007), destaca etapas particulares a fin de responder a los requerimientos suscitados con la implantación de los productos en las sucesivas cohortes. En función a la experiencia desarrollada, enuncia las etapas como:

- **Análisis de factibilidad**, de requerimientos funcionales y no funcionales del sistema, definición de limitaciones tecnológicas. Se define el producto software y se determina su factibilidad en el ciclo de vida desde la perspectiva de la relación costo-beneficio, como así también las ventajas y desventajas respecto de otros productos.

- **Especificación de requisitos del prototipo.**

Consiste en especificar las funciones requeridas, las interfaces y el rendimiento esperado para el prototipo.

- **Definición de seguridad en el acceso a la información.** Los aspectos de seguridad en el acceso y manipulación de los datos, dan lugar a establecer medidas para prevenir cualquier tipo de problemas tanto externos como internos que puedan influir en el desempeño normal de las ejecuciones.

- **Diseño del prototipo de entorno virtual.** El diseño del entorno se inicia con la definición de los requerimientos. Se contemplan: el contenido, las funcionalidades y el propósito del mismo, a quienes estará dirigido y la plataforma en la cual se implementará. Se contemplan las características como por ejemplo: i) interactividad, ii) integración de contenidos en múltiples formatos, iii) definición del objetivo de implementación. En el diseño de las interfaces se atendieron a aspectos de navegabilidad, accesibilidad y comunicación.

- **Selección de herramientas.** Los soportes tecnológicos empleados en los entornos visuales son los lenguajes de programación para construir aplicaciones web como ASP (Active Server Page) y PHP (Preprocessor HiperText), herramientas para la construcción de simuladores y autoevaluaciones en línea.

- **Desarrollo del prototipo.** En cada uno de los artefactos, se realizaron las codificaciones y simulaciones en función a las especificaciones. La integración de distintos artefactos a un mismo entorno requiere uniformar las interfaces, la comunicación entre las opciones disponibles para finalmente integrar en un único producto.

- **Recopilación e Incorporación de contenidos.**

Se utilizaron diversos formatos para la presentación de los contenidos a fin de flexibilizar y asegurar la recepción de la información al destinatario, como puede ser optar por: i) formato PDF. Las herramientas que brinda Adobe, permiten construir documentos navegables, de este modo es factible desarrollar material educativo interactivo que oriente a los estudiantes en el aprendizaje de los contenidos. ii) Formato documento de texto. Las notas tipos, parametrizables por los estudiantes, se exponen en documentos de formatos Word. iii) Acceso al material en formato comprimido. Esto con el fin de brindar la posibilidad de descargar desde la red, mediante el servicio de transferencia electrónica de archivos (FTP), las unidades temáticas y las guías de trabajos prácticos.

- **Validaciones del prototipo.** Se ejecutan pruebas de eficiencia y de robustez del código y de los simuladores. Se realizaron sesiones con usuarios dedicados a la validación del prototipo. La información obtenida es relevante como fuente de retroalimentación para el diseño tecnológico y pedagógico de los materiales didácticos en uso y aquellos previstos.

- **Refinamiento iterativo.** Permite volver a la etapa de “Especificación de requisitos del prototipo” a fin de aumentar la funcionalidad del prototipo o continuar, en función a si se logró el objetivo y los alcances esperados por los usuarios.

- **Desarrollo final.** Consiste en ajustar las restricciones o condiciones finales e integrar los últimos módulos”.

Adicionalmente, deben realizarse los procedimientos de:

- **Documentación.** Para cada desarrollo se elaboran documentaciones de soporte en el análisis, el diseño y la implementación de las soluciones de software propuestas.

- **Implementación.** La presentación de versiones, constituye un medio para obtener retroalimentación con la finalidad de refinar el entorno, de modo que al final del proyecto el resultado cubra los requerimientos. Se requiere adoptar una metodología para la integración de sistemas, siendo el principal objetivo cumplir con el concepto ciclo de vida, enfatizando el desarrollo de software y estableciendo únicamente necesidades en hardware.

- **Actualización y mantenimiento.** La actualización y/o mantenimiento, tiene razón considerando modificaciones en función a: i) nuevos requerimientos o cambios en la administración de los contenidos y/o opciones. ii) fallas detectadas por el uso en el aula.

- **Resguardo de la información.** Los datos almacenados y los códigos desarrollados deben exportarse regularmente con el propósito de asegurar el resguardo de la información.

- **Capacitación,** definido el entorno y con el objeto de propiciar el empleo de las herramientas de interactividad, se planifican instancias de capacitación *ad-hoc* orientadas a los usuarios”.

10 HERRAMIENTAS DE PROTOTIPADO

Como herramientas, los autores señalan los lenguajes dinámicos, los lenguajes de cuarta generación, y de ensamblaje y aplicaciones: pudieran ser: Smalltalk (basado en objetos, sistemas interactivos), Java (basado en objetos, sistemas interactivos); Prolog (lógico, procesamiento simbólico); LISP (basado en listas, procesamiento simbólico) y muchos otros, en

relación a la decisión del equipo de prototipado. Otros mencionan aplicaciones específicas como “Mockup, una maqueta gráfica o boceto preliminar de cómo sería el software que estamos diseñando, algunas de estas herramientas también sirven para realizar Wireframing; una ilustración visual de una página web, donde se pretenden mostrar todos los elementos que la conforman sin definir la apariencia o diseño gráfico”²⁴.

De menor a mayor formalidad, se hacen prototipos contando con:

- Papel y lápiz
- Pizarra
- Computadora
- Software de dibujo
- Aplicaciones para desarrollar demos
- Software de animación y presentaciones
- Perl + Motif + Tcl/Tk (UNIX)
- Herramientas visuales para RAD, como Visual Basic, Optima ++ y Borland Delphi
- 4GLs
- Sistemas de gestión de interfaz de usuario (UIM)
- Lenguajes de especificación ejecutable

Además, desde el punto de vista de la usabilidad, se establecen como técnicas de prototipado:

- *Borradores en papel*: Al principio del proceso de diseño se pueden crear prototipos sobre papel para mostrarlos al usuario. Normalmente son representaciones de las ventanas de la aplicación. El diseñador actúa como sistema, presentando al usuario el siguiente elemento cuando ocurre una transición entre ventanas [Nielsen93].
- *Técnica del Mago de Oz* [Preece94]: Un experto humano actúa como sistema, dando las respuestas a las peticiones del usuario. El usuario interactúa normalmente con el terminal, pero en vez de haber un software detrás, un desarrollador está frente a otro terminal conectado al primero a través de la red, y va tecleando la supuesta respuesta del sistema. El usuario normalmente no sabe acerca del “truco”, para conseguir la sensación de estar tratando con un sistema real.
- *Escenarios, storyboards y viñetas*: Un escenario describe una historia de ficción de un usuario interactuando con el sistema en una situación concreta. Las viñetas son representaciones que capturan la interacción que ocurre en un escenario. *Storyboards* son secuencias de viñetas que se centran en las principales acciones en una situación dada [Preece94]. Estas técnicas posibilitan que el equipo de diseño piense acerca de lo apropiado del diseño actual a las necesidades del usuario, favoreciendo un proceso de diseño más centrado en el usuario²⁵.

11 VENTAJAS Y DESVENTAJAS DEL PROTOTIPADO

Ventajas

La realización de los prototipos, en resumen, tiene como ventajas:

- Que con ellos se hacen las pruebas de usabilidad al principio del proceso de desarrollo.
- Incitan a la crítica necesaria de los usuarios, ya que pueden ser de bajo costo y de baja fidelidad. Si un usuario se le presenta una primera versión de un sistema que ha

requerido un trabajo sustancial, y está muy elaborado, él / ella usuario o evaluador, gerencia, es poco probable que sean más reacios para criticarlo.

- Cumplen con el deseo de mostrar resultados rápidos para el cliente
- Llamam a "tierra" la discusión durante una sesión de usuario, por lo que permite que se limite la dosis excesiva de pista, o requerimientos adicionales sobre un producto.
- Logra que el diseñador se centre en el contenido y la funcionalidad y desviar la atención de los detalles de diseño gráfico.
- Permiten obtener una aprobación formal del diseño de los programadores y el cliente antes de proceder a la etapa de desarrollo.
- Pueden ayudar a facilitar el trabajo a través de las fronteras disciplinarias, que reúne a un equipo dispar.
- Permite legalizar la experimentación, ya que son de bajo costo de alterar²⁶.

Desventajas de los Prototipos²⁷

- En un entorno de creación de prototipos de alta tecnología, tienden a ser moldeado por las herramientas que están disponibles, y para alargar el diseño, en vez de acortarlo y en lugar de responder a las necesidades de los usuarios.
- El trabajo en el prototipo puede desviar la atención de los problemas a resolver en el sistema final.
- Los desarrolladores pueden quedar atrapados en un bucle sin fin de refinamiento.
- Si el costo de implementar una idea es demasiado alta, las ideas serán rechazadas demasiado pronto en el ciclo.
- Si paralelamente deja de llevarse orden en la documentación o queda sin hacerse, luego se puede perder el razonamiento de los factores que llevan al prototipo, y las diferencias finales con el resultado.

12 CONCLUSIONES

El prototipo es un mapa, esquema, dibujo, ejemplar, código o artefacto, una muestra de un proceso, producto o servicio, que será utilizado como parte del sistema para ilustrar ciertos aspectos de su vista, las funcionalidades y clarificar los requerimientos. Un prototipo llegar a ser una representación de un sistema, ya que posee las características del sistema, sin que ello signifique su operatividad total.

Hay diferentes tipos de prototipos que dependen de cómo se utilizan o se han construido; según las distintas metodologías de desarrollo software, podemos decir que hay prototipos experimentales, exploratorios, iterativos, evolutivos, con características de baja fidelidad o alta; horizontales o verticales; orientados por la tarea o por el escenario. En los estudios de la ingeniería del software encontramos varias evaluaciones del prototipo, en cuanto a que debe ser

"desechable" o "rápido de usar y tirar", hasta llegar a ser una "muestra", pero por otra parte, otros autores desean mayor fidelidad y funcionalidad del prototipo, hasta que, gradualmente lleve al desarrollo del sistema; sin embargo, desde el punto de vista del diseño, se puede romper radicalmente con un parámetro establecido, allí radicará lo verdaderamente evolutivo, en vez desde el punto de vista iterativo, donde se fijan los cambios a partir de los requisitos.

El usuario prueba algo, ve lo que sucede, luego lo modifica, esta interacción proporciona una retroalimentación instantánea y le permite ver al usuario resultados y sugerir modificaciones para su culminación, vinculada a la satisfacción de la necesidad y por ende el alcance de los objetivos. Construir prototipo tiene sus ventajas, hacia la mejor satisfacción del usuario y desventajas, ya que conlleva desafíos al ingeniero y al equipo, sin embargo, se pudieran ofrecer cosas que luego en el sistema ni serán operativas, ni funcionales; por otra parte, hacer el esfuerzo de construirlo puede ayudar a refinar los razonamientos y llegar con menor esfuerzo a soluciones eficaces.

De la Autora



Emma Corina Salazar Suárez. Socióloga de la Universidad Católica Andrés Bello (UCAB, 2001); Orientadora IRFA (1995) y facilitadora para algunas instituciones como el Centro Gumilla (1998), a nivel presencial y virtual (2010); tiene estudios diversos y certificados en varias áreas en promoción de la salud (1995, 2003), levantamiento y evaluación de la calidad de la data y documentación de la gestión de proyectos y programas (2003-2010) en organismos públicos; genera y realiza análisis de contenidos (2012). Autodidacta en Ecología, tiene en su haber diversos cursos desde básico para el reconocimiento de árboles; Introducción a la Economía Ecológica (2007), con ECOPORTAL-GEPAMA, y participante de varios talleres para evaluar calidad del agua PAVCO (2008); Petróleo y realidades energéticas (2009); del Aula Abierta sobre Cambio Climático hacia las comunidades de la UCV (2012) y del Plan Metropolitano. Lleva más de 7 años en defensa y cuidado del ambiente, a través de la activación en la difusión del desarrollo sustentable en Venezuela. Actualmente, cursando el postgrado en Sistemas de Información en la UCAB. Se desempeña en libre ejercicio, en la obtención y procesamiento de data, en estudios sobre opinión pública, mediante la investigación acción, así como evaluando la pertinencia y usabilidad de sistemas y páginas web por parte de los usuarios. Divulga sus observaciones, organiza eventos, conversatorios, vinculados a efemérides ecológicas, dicta cursos y talleres; difunde información, experiencias y conocimiento, opina sobre la situación de su país, en varios blogs, y en especial promueve la educación ambiental ciudadana y popular en: www.ecorina.blogspot.com.

Artículo elaborado para la materia Ingeniería del Software a cargo del Dr. Pedro Bonillo. Caracas, Venezuela, Diciembre 2012

13 REFERENCIAS

- ¹ Martínez Ron, Antonio, 2009. *Diez maravillas tecnológicas de la antigüedad*. Fogonazos Asombros Diarios. Disponible en: <http://www.fogonazos.es/2009/05/diez-maravillas-tecnologicas-de-la.html>. Consultado 01/12/2012.
- ² Diccionario Etimológico, 2012. Origen de la palabra Prototipo. Consultado 01/12/12. Disponible en: <http://etimologias.dechile.net/?prototipo>
- ³ Porcel Granados, Francisco, 2009. *Desarrollo Tecnológico en la Historia de la Humanidad: Inventores e Inventos, Zeppelin y el dirigible*. Revista Digital, Innovación y Experiencias Educativas. 2009. Disponible en: http://www.csi-csif.es/andalucia/modules/mod_ense/revista/pdf/Numero_20/FRANCISCO%20M_%20PORCEL%20GRANADOS_2.pdf, Pág. 2.
- ⁴ Machado La Torre. 2000. Especialidad de Postgrado en Cómputo Aplicado, ISEI, Colegio de Postgraduados <http://colposfesz.galeon.com/hiacomp/primerag.htm>
- ⁵ Diccionario de la Real Academia Española de la Lengua. Prototipo. Consultado 04/12/2012. Disponible en: <http://lema.rae.es/drae/?val=prototipo>
- ⁶ Farxel, The Free dictionary, 2012. Prototipo. Basado en Manual Diccionario de la Lengua Española, 2007. Consultado 03/12/2012 Disponible en: <http://es.thefreedictionary.com/prototipo/> También en: <http://www.wordreference.com/definicion/prototipo>
- ⁷ Maner, Walter, 1997. Prototipado. Traducción original, Universidad de Zaragoza, 2006. Consultado el 27/10/2012. Disponible en: <http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm>
- ⁸ Diccionario Etimológico, 2012. Origen de la palabra Ingeniero. Consultado 01/12/2012. Disponible en: <http://etimologias.dechile.net/?ingeniero>.
- ⁹ Carr, M., Verner, J. *Prototyping and Software Development Approaches*. Consultado 02/12/2012 <http://www.cb.cityu.edu.hk/is/getFileWorkingPaper.cfm?id=55>. Traducción propia asistida con recursos de traducción en Web.
- ¹⁰ Alvear, Freddy. Modelos, simuladores y prototipos, 2007. Consultado 05/12/2012. Disponible en: <http://prototipod.blogspot.com/2007/04/modelos-simuladores-y-prototipos.html>
- ¹¹ Reyes, Leydi, María Ruiz y Vivancos, 2009. Prototipos Informáticos. Consultado 30/10/2012. Disponible en: <http://sistemas2009unl.wordpress.com/prototipos-informaticos/>
- ¹² Ursom, JM y Austin, John Langshaw, 1962. *Cómo hacer cosas con palabras.: Palabras y acciones (How to Do Things with Words)*. Barcelona: Paidós, 1982. (traducción de la edición original inglesa por JacquesDerrida). Disponible en: http://www.jacquesderrida.com.ar/restos/hacer_cosas_palabras.pdf
- ¹³ Universidad de Murcia, 2011. *Informática Aplicada a la Gestión Pública*. Facultad Derecho. Capítulo 1. Ingeniería del software. Introducción. Disponible en <http://www.um.es/docencia/barzana/IAGP/IAGP2-Ingenieria-software-introduccion.html>. Consultado 05/12/2012.
- ¹⁴ Bonillo, Pedro. *Diplomado en Software Libre, Modulo. 3 Desarrollo Tecnológico; 3.2 Ingeniería del Software*. Consultado 30/10/12. Disponible en http://colabora.softwarelibre.gob.ve/home/IS_DiplomadoSWLv2-1.pdf
- ¹⁵ Instituto de Ingenieros Eléctricos y Electrónicos, 2012. Historia de la IEEE. Consultado 03/12/2012. Disponible en: http://www.ieee.org/about/ieee_history.html
- ¹⁶ Universidad de Murcia. *Capítulo 1. Ingeniería del software. Introducción* <http://www.um.es/docencia/barzana/IAGP/IAGP2-Ingenieria-software-introduccion.html>
- ¹⁷ Pereyra, German. El uso del prototipo en el desarrollo del sistema. Consultado 05/12/2012 Disponible en: <http://www.monografias.com/trabajos12/proto/proto.shtml>
- ¹⁸ Mata Gómez, Ipswich Nir. 2011. *Infografía sobre Metodología de Desarrollo de Sistema por Prototipos*. Material para la Cátedra. Inédito. 5págs.
- ¹⁹ Floyd, C. 1984. *A Systematic look at prototyping*. Consultado 05/12/2012. Disponible en: <http://laerer.rhs.dk/susaneru/prototyping/floyd-prototyping.pdf>
- ²⁰ Bosse Westerlund. Ptototipo. Ejemplos. Consultado 05/12/2012. Disponible en: <http://www.csc.kth.se/utbildning/kth/kurser/DH2622/mdifk07/prototyper-bw-mdifk.pdf>
- ²¹ Rivas, Daniel E., Poletti, Leopoldo A. y Salinas, Luis G. *Una Herramienta de Soporte a la Ingeniería de Requerimientos "Prototyping of Web Applications (PWA)" 2012*. Facultad Politécnica de la Universidad Nacional de Asunción. Disponible en:
- ²² Meza Carlos y Barcanegras. SF. *Tipos de prototipos*. Presentación 21 paginas. Disponible en: <http://www.slideshare.net/walter4154/prototipos#btnNext>
- ²³ Piattini, Mario, Jose A. Calvo Manzano, Joaquín Cervera y Luis Fernández. 2000. Análisis y diseño detallado de Aplicaciones informáticas de Gestión. Alfa_Omega, rama.Págs 164 a la 166.
- ²⁴ Herramientas para realizar prototipos. Consultado 01/12/2012: Disponible en: <http://www.awven.com/q216-herramientas-realizar-prototipos-mockups-wireframing/>
- ²⁵ Ferre Grau, Xavier. 1999. *Principios Básicos de Usabilidad para Ingenieros Software*. Disponible en: <http://educacion.usach.cl/ojs/index.php/ojsprueba1/article/viewFile/7/5/> También citado en Técnicas de usabilidad. Estudio exploratorio sobre su incorporación en los procesos de desarrollo software en pymes locales. 2006. Argentina, en: <http://www.frre.utn.edu.ar/llJCyT/clean/files/get/item/2196>
- ²⁶ Basado en Soegaard, Mads, 2010. *Prototyping*. Consultado 4 de Diciembre de 2012. Disponible en: <http://www.interaction-design.org/encyclopedia/prototyping.html>. Orden propio de las ventajas señaladas.
- ²⁷ Wilson, Diane, Rauch Thyra, y Paige Joeann , 1992. *Prototipos y el ciclo de desarrollo de software*. Consultado 05/12/2012. Disponible en: <http://www.firelily.com/opinions/cycle.html>