General

Descripción conceptual

Una aplicación monolítica describe una aplicación de software de un solo nivel en la que la interfaz de usuario y el código de acceso a datos se combinan en un solo programa desde una sola plataforma, es decir, está diseñada sin modularidad.

Una aplicación monolítica en si es independiente de otras aplicaciones informáticas, la cual se estructura en grupos funcionales muy acoplados, involucrando los aspectos referidos a la presentación, procesamiento y almacenamiento de la información. Son implementadas dentro de un solo componente de software.

La filosofía de diseño es que la aplicación es responsable, no solo de una tarea en particular, sino que puede realizar cada paso necesario para completar una función concreta.

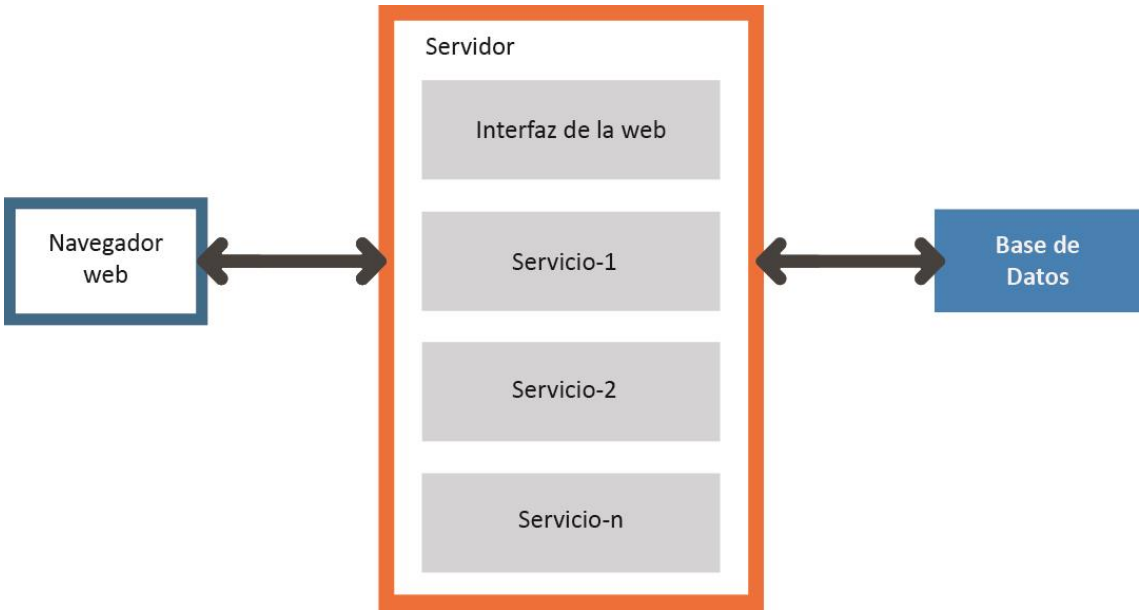

Figura 1. Estructura de una aplicación monolítica

Otras características de las aplicaciones monolíticas es que la construcción del programa final se basa en componentes compilados al mismo tiempo. También hay que tener en cuenta que carecen de protecciones y privilegios al entrar en rutinas que manejan diferentes aspectos de los recursos del computador, como memoria, disco, etc.

Un alto número de aplicaciones monolíticas están hechas a medida, por lo que son eficientes y rápidas en su ejecución y gestión, pero como consecuencia, carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones. Es decir, si más tarde se quiere ampliar los servicios que ofrecen puede llegar a resultar un verdadero problema de desarrollo.

Características principales

Conceptos tecnológicos subyacentes

Aparición y desarrollo histórico

Históricamente, representan la estructura de los primeros softwares, constituidos fundamentalmente por un solo programa compuesto por un conjunto de funciones entrelazadas de tal forma que cada una puede llamar a cualquier otra.

En su uso original, el término "monolítico" describía enormes aplicaciones *mainframe*[1] sin modularidad utilizable. Esto, en combinación con un rápido aumento en el poder computacional y, por lo tanto, un aumento rápido en la **complejidad de los problemas** que podrían ser abordados por el software, dio como resultado sistemas no sostenibles y la "**crisis del software**".

Crisis del Software

Este término se refiere a los diversos problemas que han experimentado, desde sus inicios, las empresas dedicadas al desarrollo del software.

Estos problemas van desde la incapacidad de la propia empresa a desarrollar un software con la calidad esperad, hasta causas derivadas de los continuos cambios en las funcionalidades por parte del cliente, para la aplicación que desea.

En 1968, durante la primera conferencia elaborada por la OTAN[2], Friedrich L. Bauer habló por primera vez del conjunto de dificultades y/o errores ocurridos en la planificación, estimación del coste, productividad y calidad de un software, es decir, de lo que se conoce como la "**crisis del software**". Dicho término se le atribuyó, aunque ya había sido utilizado por Edsger Dijkstra en su libro "The Humble Programmer".

Para dar solución a los problemas que se presentaban, en esta conferencia se creó una nueva rama de ingeniería, la **ingeniería del software**[3].

Causas

Como se ha introducido anteriormente, existen múltiples causas que originan la **crisis del software**.

La primera de ellas a considerar es que el desarrollo de software es un proceso relativamente nuevo, por lo que no se cuenta en muchas ocasiones con personal lo suficientemente capacitado para ello.

En los años 60s se empezó a notar que las técnicas usadas para programar habían quedado obsoletas. Aun así, algunas personas todavía creían que la programación de software debía considerarse un arte, una actividad más creativa que disciplinada. En aquella época, otro de los problemas existentes es que muchos programadores no tenían una formación enfocada al desarrollo, sino que habían aprendido por ellos mismos. Esto provocaba que los resultados se obtuvieran pasada la fecha de entrega, los programas no funcionaban de manera correcta, era difícil realizar cambios y se excedían los plazos y costes planeados.

En 1981, Barry Boehm estimó que el coste total en software en Estados Unidos fue del 2% del **Producto Interno Bruto** de 1980, lo que representa $40 billones. En 1985 el coste total se elevó a $70 billones en Estados Unidos y $140 billones en el resto del mundo. Para 1999 Boehm y Sullivan estimaron que el coste en 1998 aumentó a los $300 - 400 billones en Estados Unidos y casi el doble a nivel mundial.

El coste total del software es una parte muy importante del desarrollo, ya que no solo involucra la ejecución del mismo, sino que también añade el coste de mantenimiento tras la entrega. Es aquí donde se pueden encontrar la mayor concentración de problemas ya que, generalmente, los proyectos carecían de una planificación previa y se trabajaba sobre la marcha.

Los errores más comunes que afectan a lámala planificación desde la empresa son:

- mala redacción del código (38.33%),
- diseño incorrecto (24.17%)
- mala documentación (13.33%)
- errores de requerimientos (12.50%)
- correcciones mal implementadas (11.67%)

Por otro lado, dado que el software es un conjunto de programas/instrucciones de un computador, y que por lo tanto, no es un elemento de carácter físico; es poco probable que resulte exitoso el primer intento de elaboración, ya que el equipo encargado no posee la misma visión de los requisitos que tiene su cliente, lo que a su vez complica el diseño al detalle de estos.

En algunas ocasiones, el propio cliente es el que no tiene claras las características del producto que desea, o tras el análisis de requisitos y comienzo del desarrollo, añade nuevos que modifican en menor o mayor medida lo implementado hasta el momento. Este hecho hace que, el desarrollo de aplicaciones monolíticas se resienta ya que el cliente necesita un producto ajustado a sus requerimientos.

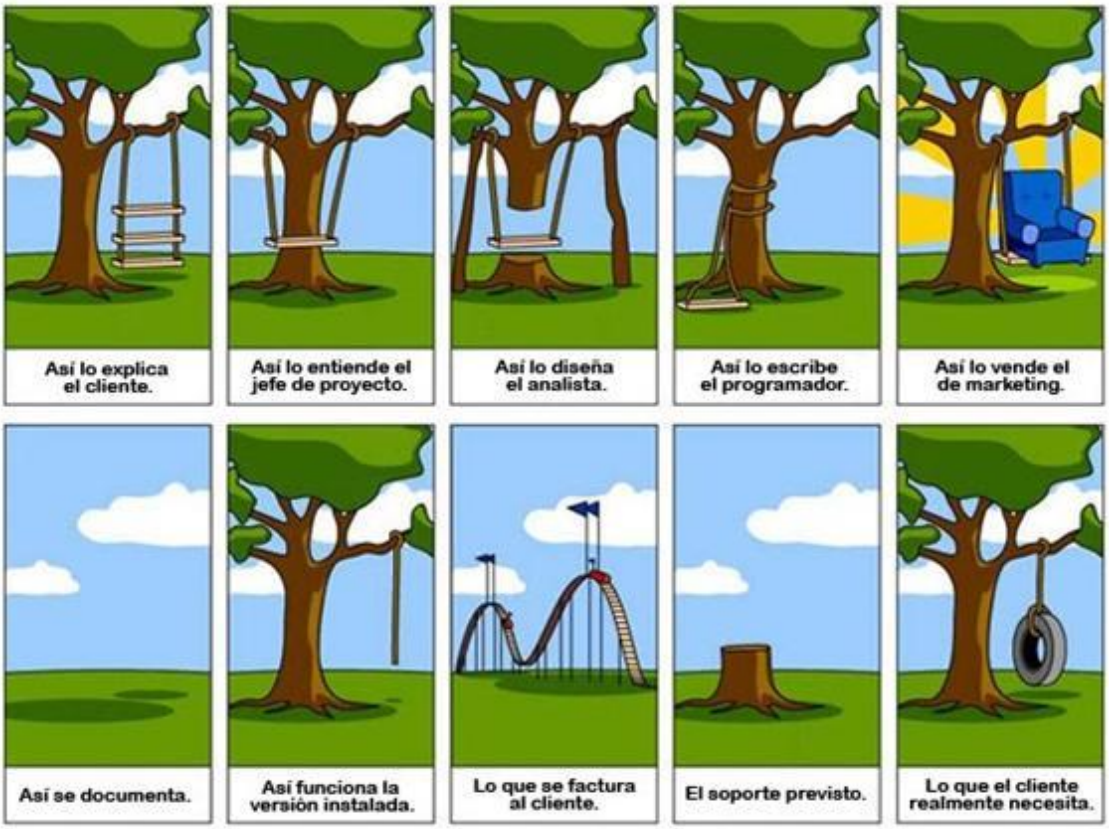

Figura 2. Problemática del desarrollo del software

Desarrollo de aplicaciones web

En cuanto a lo que a aplicaciones web se refiere, no fue hasta la década de los 90 cuando saltó al plano doméstico gracias a que la World Wide Web[4] ganó gran aceptación con la introducción del navegador Mosaic[5] en 1993 y poco después, las empresas comenzaron a reconocer el potencial comercial de la web. La infraestructura de red creció para acomodar lo que demostraría ser una afluencia masiva de actividad en línea.

En 1989, Tim Berners-Lee (entonces miembro del CERN[6]) describió su concepto de una plataforma informática que podría facilitar la colaboración entre investigadores que se encuentran en diferentes partes del mundo. Esto condujo a la invención del HTML[7], en 1990. Este lenguaje se convirtió en el componente fundamental de la World Wide Web y permanece en el núcleo de su codificación e infraestructura. El estándar habilitó codificadores con la capacidad de organizar diseños de página web que podrían ser entendidos e interactuados con redes interconectadas.

La web avanzó mucho en los años posteriores a la crisis tecnológica de 2000-2001. Durante este tiempo, los gobiernos comenzaron a desempeñar papeles cada vez más influyentes en la web, mientras que, al mismo tiempo, las empresas de tecnología resurgieron tras el colapso para establecer un nuevo curso en el comercio y la cultura, digitalizándolas. Y a medida que se estableció esta base más nueva y sólida, Internet se convirtió cada vez más en el principal

canal de telecomunicaciones en la era moderna.

A medida que las mejoras de hardware cultivaron redes más amplias y un mayor ancho de banda, el desarrollo web también lo hizo permitiendo a los diseñadores una variedad de multimedia a incorporar en la presentación web.

Referencias

- **Friedrich L. Bauer** (1924 -2015): fue un ingeniero informático nacido en Alemania, profesor en la Universidad Técnica de Munich.

- **Edsger Dijkstra** (1930-2002): fue un científico de la computación nacido en los Países Bajos y uno de los mayores desarrolladores del software de su época.

- **Barry Boehm**: ingeniero informático estadounidense, profesor emérito de esta materia en el Dpto. de ciencias tecnológicas en la Universidad del Sur de California.

- **Boehm y Sullivan**: ingenieros informáticos nacidos en Estados unidos. Boehm es profesor emérito de esta materia en el departamento de ciencias tecnológicas en la Universidad del Sur de California, y Sullivan en la Universidad de Virginia.

- **Tim Berners-Lee**: científico de la computación nacido en Reino Unido, conocido por ser el padre de la Web y establecer la primera comunicación entre un cliente y un servidor usando el protocolo HTTP (noviembre de 1989)

Glosario terminológico

[1] **mainframe**: son computadores utilizadas principalmente por grandes organizaciones para aplicaciones críticas; procesamiento de datos a granel, como censos, estadísticas de la industria y del consumidor; planificación de recursos empresariales y procesamiento de transacciones.

[2] **OTAN**: Organización del Tratado del Atlántico Norte, alianza militar intergubernamental basada en el Tratado del Atlántico , firmado el 4 de abril de 1949. La organización constituye un sistema de defensa colectiva, en la cual los Estados miembros acuerdan defender a cualquiera de sus miembros si son atacados por una facción externa.

[3] **Ingeniería del software**: una disciplina que integra métodos, herramientas y procedimientos para el desarrollo de SW de computador.

[4] **World Wide Web**: sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet

[5] **Mosaic**: fue el primer navegador web gráfico disponible para visualizar páginas web en el sistema operativo Microsoft Windows.

[6] **CERN**: Organización Europea para la Investigación Nuclear, es el mayor laboratorio de investigación en física de partículas del mundo

[7] **HTML**: Lenguaje de marcado de hipertexto, es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, definiendo una estructura básica y un código para la definición de contenidos.

Bibliografía

1. Wikipedia, https://en.wikipedia.org (fecha acceso: 17/11/2017)
2. Blog Historia de la informática, https://histinf.blogs.upv.es/category/historia/ (fecha acceso: 25/11/2017)
3. Techopedia, https://www.techopedia.com (fecha acceso: 22/11/2017)