Diseño de Interfaces

Edgar Quiroz

26 de julio de 2021

Índice

1. Interacciones Humano-Máquina

Estudia el diseño, evaluación e implementación de sistemas computacionales interactivos. En otras palabras, mecanismos para comu-

nicarse con una máquina.

Interacción: acción recíproca entre dos o más agentes

Gran parte del éxito de un sistema está dado por que tan **seguro**, **útil**, **eficaz** y **usable** es. Esto toma lugar en **interfaces**.

- **Seguro**: robusto a ataques
- **Útil**:
- **Eficaz**: hace el software productivo y ahorra costos de capacitación.
- **Úsabilidad**: se sencillo e intuitivo.

Tarea: diseñar reglas para jugar serpientes y escaleras usando únicamente texto.

Para llegar a mecanismos con estas características, en general se estudia como interactúan las personas interractúan con su ambiente y con otras personas, e intentar replicarla.

1.1. Algo de historia

Un ejemplo sería la **ENIAC**. Ésta se configuraba con más de 6000 interruptores. Para usarla había que tener un conocimiento muy profundo de la aquitectura de la máquina, y modificaciones a su funcionamiento tomaba semanas.

Una mejora a esto fue la introducción de **teletipos** (con tarjetas perforadas). Esto facilitaba la forma de modificar el software de la máquina. En 1968, Xerox creó la primera **interfaz gráfica**. Poseía un mouse, ventanas, íconos y menúes.

2. UML

En cualquier intento colaborativo para desarrollar software debe haber una forma de **especificar** que se va a hacer. Hay dos casos extremos potencialmente problemáticos. Primero en **equipos grandes** y multidisciplinario. Aquí la comunicación directa no es práctica, y las diferentes áreas de todos no permiten usar lenguaje

técnico. El otro caso sería cuando hay un **proyecto grande**, tal que no es factible que todos los miembros del equipo entiendan todo el sistema.

Para resolver esto se creó **UML** (Universal Modelling Language). Es un estándar de decenas de diferentes tipos de diagramas, cada uno de ellos especializado en **especificar un aspecto** del sistema de software de manera **simple y no técnica**

Nota: al iniciar un trabajo en una empresa de software, pedir toda la documentación disponible para entender como funciona el proyecto donde vas a trabajar.

2.1. Diagramas de clases

Permiten modelar parte del sistema de software usando **orientación a objetos**. En breve, diferentes elementos y actores se agrupan en **clases**. Cada clase tiene ciertas caracterísicas propias, y puede interactuar con las otras clases a través de **métodos**.

2.2. Diagramas de flujo

Permiten modelar un **algoritmo**. Se pueden tomar decisiones, asignar variables y definir ciclos. Estos algoritmos normalmente reflejan un **proceso** que se quiere automatizar en el sistema.

Todos los sistemas tienen un flujo de actividades. A grandes rasgos siempre es útil modelar el flujo, sin importar el tamaño del proyecto.

2.3. Diagrama de secuencia

Describe **como y en que orden** interactuan las diferentes entidades que participan en un flujo. Pueden incluir tanto flujos **exitosos**, flujos **fallidos** o flujos **excepcionales**.

2.4. Diagrama de despliegue

Describe la **implementación física** del sistema, i.e. los servidores, y como estén conectados entre sí. Si se usa un servicio de aloja-

miento, generalmente tiene herramientas para crear estos diagramas, como es el caso de **AWS**.

2.5. Diagramas de casos de uso

Describe la **funcionalidad específica** (paso a paso) de un sistema para los diferentes **tipos de usuario**. Tener bien identificados todos las funcionalidades permite crear interfaces más especializadas.

Determinar las funcionalidades requiere un **análisis de requerimientos**.

Nota: el análisis de requerimientos es un buen punto de partida para diseñar una interfaz, aunque es algo bastante posterior.

3. Elementos de una interfaz gráfica

Los usuarios van a **interactuar** con la interfaz gráfica para realizar alguna **tarea**. Esto implica **usuarios**, **contenido gráfico** y **tareas objetivo**.

3.1. Usuarios

Tiene edad, escolaridad, conocimiento del tema a presentar, estilo de aprendizaje, nacionalidad, contexto socio-cultural, experiencia con equipos de cómputo, discapacides.

Los usuarios necesitan realizar cientas tareas en la interfaz a diseñar. Definirlas claramente y hacer **pruebas de usabilidad** constante es importate.

3.1.1. Artefactos de personas

Usuario **ficticio** basado en datos de usuarios reales. Normalmente, se **segmentan** los usuarios en base a caracteristicas comunes. A cada segmento se le asocian dos o tres artefactos de personas. De forma realista, las personas no suelen usar toda la funcionalidad del sistema. Esta técnica no puede reemplazar interacción con usuarios **reales**.

Algunos datos útiles podrían ser

- Fotografía
- Demografía: edad, puesto de trabajo, residencia, estado civil
- Disponibilidad de tecnología
- Motivaciones: hobbies
- Limitaciones: discapacidades

Al realizar cambios a la interfaz, hay que tener en cuenta como afectaría a estas personas ficticias.

3.1.2. Diagramas de empatía

Describen las **emociones** de los usuarios en seis categorías. Estos usuarios suelen ser artefactos de persona.

- Vista: cuál es su cotiniadidad, su círculo social
- Oído: que lo influencia, ídolos, marcas favoritas, que productos consume
- Pensamientos: preocupaciones, sueños, autopercepción
- Habla: hobbies, personalidad
- Dolor: miedos, frustraciones, obstáculos
- Necesidad: concepto del éxito, objetivo, solución a sus problemas

Después de tener los mapas de empatía para los artefactos, es útil **validarlos** contra dato estadísticos de usuarios reales.

3.2. Contenido

Se debe tomar en cuenta el **propósito** de la interfaz. Entre más **restrigido** sea, más **sencillo** debería ser el diseño. Además de esto,se debe definir el formato y **cantidad de información** a mostrar, favoreciendo lo más breve.

3.2.1. Navegación

Representa el mapa de las **posibles pantallas** de la interfaz y a dónde pueden llevar. Permiten ver la **organización de la información** y detectar más facilmente posibles problemas estrucuturales, como **desbalance**. Cuando los proyectos están enfocados a **funcionalidades** (esto es más típico de las Intranets y Extranets) se suele usar un **diagrama de interacción** como sustituto.

Algunos tipos son:

- Lineal: solo página siguiente y página anterior. Útil cuando se tiene una única tarea.
- Estrella: solo página inicial que puede acceder y es accesible desde todas las demás
- Arborescente: árbol definido por una jerarquía de información
- Múltiple: todos accesibles desde todos
- Red: se puede navegar entre temas similares sin restricciones estrucuturales
- Metafórica: simulaciones de espacios reales usano simbolos familiares.

4. Evaluación y consideraciones

4.1. Generalidades de usabilidad

Segun la **ISO/IEC9241-11**, es una medidad sobre que tan útil es un producto de software para que cierto grupo de usuarios para realizar actividades especificos de forma eficaz. Midiendo estas caracterísiticas se pueden medir la usabilidad de un sistema.

Es diferente de parte de marketing, ergonomía o accesibilidad. Debe fácil de usar, productivo, cumple las metas funcionales. De forma intuitiva, es la «calidad» del producto de software. Se logra con una mezcla de buen diseño y buen funcionamiento.

4.2. Generalidades para diseñar

Al diseñar el sistema, se debe adaptar al usuario, ocultando las complejidades del sistema. Esto se hace por medio de interfaces gráficas, auditivas, tactil, movimiento, navegación.

Hay que perfilar el tipo de usuario. Para esto se suelen usar herramientas como los arquetipos de usuario o los mapas de empatía (a revisar a fondo después).

Luego, definir las tareas a realizar (**historias de usuario**) y que se necesita para hacerlo. Es decir, entender lo que los usuarios quieren, observar los lugares de trabajo, indentificar sus necesidades (tanto como las que te dan explícitamente como las que tú identificaste) y diseñar soluciones a estas necesidades. Esto se conoce como principio de **diseño centrado en usuario**.

Además de esto, hay que revisar los posibles contextos, como el tipo de equipo donde se usará el sistema, conexión, el lugar físico.

Algunos consejos al respecto son

- Poner exactamente lo necesario, ni menos ni más (incluyendo multimedia)
- No dejar de lado la legilibilidad
- No dejar de lado la navegación
- Mantener consistencia visual
- Facilitar la instalación/carga

Algunos problemas comunes pueden ser

- Priorizar el diseño o el funcionamiento
- **Iniciar etapas sin tener toda la información**
- Solo un área tome las decisiones
- Usar tecnologías nuevas
- **Intentar generaizar el diseño mas de lo necesario**
- **No ser flexible a las necesidades**

- Los directivos tiene la última palabra

Algunos consejos para evaluar el diseño pueden ser

- Muestra representativa de usuarios
- Tareas breves y específicas
- Ambiente controlado (laoboratorio de usabilidad)
- Hacer pruebas de usabilidad y pruebas de *experiencia*

Si se logra un buen diseño, se tendrá como corolario que

- No habrá secciones meramente estéticas
- Habrá pocos usuarios que decerten
- Los usuarios estarán satisfechos
- Se adoptará el producto rápidamente
- Se podrá mandar un mensaje emocional a los usuarios

Tarea: leer artículo, probar aplicación y proponer mejoras.