

Examen Web Apunte Final

Tema 1 - Intro

- Web App
 - A web application is any application that uses a web browser as a client
 - In a server-based internet application, the application uses the internet protocol to receive request from a client, typically a Web Browser, process the associated code and returns data to the browser
 - A web application (Web App) is an application program that is stored on a remote server and delivered over the internet through a browser interface
 - Es decir los datos o los archivos en los que trabajas son procesados y almacenados dentro de la web
 - Estas aplicaciones, por lo general, no necesitan ser instaladas en un ordenador
 - Las aplicaciones web son un tipo de software que se codifica en un lenguaje soportado por los navegadores web y cuya ejecución es llevada a cabo por el navegador en internet o de una intranet
 - Sistemas web
 - infraestructuras o sistemas que permiten funcionar a una aplicación web
 - Características
 - Evolución rápida
 - A gran escala y fuertemente distribuido
 - Multi usuario, lenguaje, cultural
 - Seguridad y confidencialidad
 - Diferentes medios de acceso y agentes de usuario
 - Gran volumen de información, varios formatos y procesos
 - Ventajas Web App
 - Compatibilidad multiplataforma
 - Acceso desde internet
 - Aplicaciones ligeras
 - Faciles de actualizar
 - Acceso inmediato
 - Compatibilidad de versiones
 - Ahorra tiempo de instalación y despliegue
 - No hay necesidad de instalar
 - Desventajas Web App

- Menor funcionalidad que las aplicaciones de escritorio
- Limitaciones del navegador
- Disponibilidad supeditada al proveedor de red
- Alta dependencia del servidor
- Limitación por el protocolo HTTP utilizado
- Cuello de botella en el ancho de banda para grandes cantidades de datos
- Necesidad de encriptar los datos
- Necesidad de SW adicional o versiones específicas de navegadores

- **Arquitectura**

- Cliente-Servidor
 - Servicio o funciones que requieren acceso a información difícil de distribuir equipamiento especial o capacidad de cómputo
 - Servidores, que ejecutan los servicios bajo petición
 - Clientes, que piden la ejecución de un servicio, con capacidad de cómputo reducida, y presentan la respuesta
 - Protocolo de petición - respuesta
- Capas
 - Capa de acceso
 - capa del navegador
 - Capa del servidor
 - capa de persistencia

- Sitio/página web vs Web App vs servicio web

- Sitio Web
 - Website
 - conjunto de páginas web en un mismo dominio
 - Páginas informativas de negocio
 - Requieren poca o nula interacción por parte del usuario
- Aplicación web o Web App
 - Suelen ser el propio negocio
 - Requieren interacción por parte del usuario y normalmente registrarse
 - Diseñada para humanos
 - Tiene interfaces
 - Human to machine
- Servicio web

- Diseñados para máquinas
- Exponen APIs
- Machine to Machine
- Tienen aplicaciones
- Información sin presentación
- Front y Back End
 - Front End
 - Lo que se ve
 - Web Performance
 - suavidad e interactividad
 - Conseguir que el sitio sea usable lo antes posible
 - Responsive web design
 - sirve para todos los dispositivos
 - Cross - browser compatibility
 - comprobar en diferentes navegadores
 - funcionalidad
 - rendimiento
 - accesibilidad
 - End-to-end testing
 - Replicar escenarios reales
 - Validación de integración
 - Validación de integridad de datos
 - Build automations
 - Empaquetado
 - Accessibility
 - Lograr que las páginas sean usadas por el mayor número de personas independientemente de sus habilidades
 - Usability
 - Mejora la experiencia de usuario
 - Evitar ambigüedades
 - Colocación de los elementos
 - Calidad de la experiencia
 - Technologies
 - HTML
 - CSS
 - JS
 - Back End

- Parte del servidor
- Lo que no se ve pero esta ahi
- Automated testing
 - Comprobación del servidor
 - Comprobación de las APIs
 - Comprobación de las bases de datos
 - ACID
 - CRUD
 - Schema
- Application Data Access
 - Roles
 - Seguridad
- Application Business Logic
 - Domain Logic
 - Reglas
- Database Administration
 - Funcionamiento eficiente de la DB
 - Gestión de recursos
 - Memoria
 - Disco
 - Red
- Scalability
 - Diseño pensado en la escalabilidad
 - Control y de consumo y rendimiento
- High Availability
 - Despliegue en diferentes regiones
 - Bajo tiempo de recuperación
- Security
- Software architecture
 - diseñar el sistema
 - Autenticación
 - autorización
- Back ups
- Technologies
 - Scripting
 - PHP
 - JavaScript / Node.js
 - Bases de datos

Tema 2 - HTTP

- HTTP = Hypertext Transfer Protocol
 - Protocolo de transferencia de Hipertexto
 - Capa de aplicación
 - Transmitir documentos hipermedia como HTML
 - Comunicación entre clientes y servidores
 - Protocolo de transacciones
 - Petición - respuesta
 - Protocolo sin estado
 - stateless
 - Se suele usar con TCP/IP
- Como funciona?
 - El cliente abre una conexión
 - El cliente inicia o manda una petición HTTP (REQUEST)
 - El servidor devuelve la respuesta HTTP (RESPONSE)
 - Se cierra la conexión
- Conceptos
 - User-agent
 - Agente de usuario
 - Programa que representa una persona, el cliente (pej. el navegador)
 - Stateless
 - HTTP es un protocolo sin estado
 - No se guarda información sobre conexiones anteriores
 - La respuesta del servidor es la misma para un cliente previo que para uno nuevo
 - Ventaja
 - Escalabilidad
 - Menos complejidad
 - Mayor rendimiento
 - Desventaja
 - Complica la interacción con el usuario
 - Hace falta información extra para mantener la sesión
 - Susceptible a ataques como DDoS (Distributed Denial of Services)
 - URL
 - Uniform Resource Locator
 - Localizador de Recursos uniforme

- Recurso (resource)
 - Información que solicita el cliente
 - Por ejemplo
 - Documento HTML
 - Imagen
 - Video
 - Referencia a recurso web especificando la localización en una red y un mecanismo para obtenerlo
 - Son únicas, cada URL identifica unívocamente un recurso
- Tipos de UR
 - URI
 - Uniform Resource Identifier
 - RFC
 - Schema
 - sirve para acceder a un recurso físico o abstracto por Internet.
 - URL
 - Uniform Resource Locator
 - Localizador de Recursos uniforme
 - es la dirección específica que se asigna a cada uno de los recursos disponibles en la red
 - URN
 - Uniform Resource Name
 - identificar claramente distintos tipos de objetos, algo especialmente útil en Internet, donde cada día interactúan millones de personas y aplicaciones –que también suelen intercambiar infinidad de recursos.
 - URC
 - Uniform Resource characteristic
 - Metadatos
 - Cualquier dato descriptivo que pueda guardarse sobre un recurso, que puede servir, entre otras cosas, para catalogarlo.
- Método
 - También llamado verbos
 - Acción que deseamos efectuar sobre el recurso indicado en la petición
 - Recurso podría ser un archivo que reside en un servidor, o podría ser un programa que se esta ejecutando en dicho servidor
 - SIEMPRE VAN EN MAYÚSCULAS
 - Tipos
 - GET

- obtener un recurso
- No modifica nada en el servidor
- Método obligatorio
- No tiene cuerpo
- POST
 - Envía datos al servidor para que sean procesados por el recurso especificado en la petición
 - Los datos se incluyen en el cuerpo de la petición
 - Podría crear un nuevo recurso en el servidor o actualizar un recurso ya existente
- HEAD
 - Obtener la cabecera de una petición GET sin el contenido
 - Obtener la meta información del recurso
 - Para conocer el tamaño/ version
 - Método obligatorio
- PUT
 - Envía un recurso (un archivo) al servidor
 - A diferencia de POST este método crea una nueva conexión socket y la emplea para enviar el recurso, lo cual resulta más eficiente que enviarlo dentro del cuerpo del mensaje
- DELETE
 - Elimina el recurso especificado
- TRACE
 - Pide al servidor que le envíe un mensaje de respuesta
 - Se suele emplear para diagnosticar posibles problemas de conexión
- OPTIONS
 - Pide al servidor que le indique los métodos HTTP que soporta para una determinada URL
- CONNECT
 - Se emplea para transformar una conexión ya existente a una conexión encriptada (HTTPS)
- PATCH
 - Modificar parcialmente un recurso ya existente en el servidor
- Respuestas HTTP - Códigos de estado
 - Números de tres dígitos
 - Forman parte de la respuestas HTTP
 - Explican que ha sucedido al intentar llevar a cabo una petición

- Varias Categorías
 - 1xx Mensaje
 - 2xx Operación Realizada con éxito
 - 3xx Redirección
 - 4xx Error por parte del cliente
 - 5xx Error del servidor

Tema 3 - HTML

- HTML
 - HyperText Markup Language
 - Sistema para anotar documentos
 - Distinguible del texto (contenido)
 - Formatear texto
 - Importa el orden

Tema 4 - CSS

- CSS
 - Cascading Style Sheet
 - hojas de estilo en cascada
 - Describen la presentación del documento
 - Tres tipos de CSS
 - Archivo externo
 - Archivo interno
 - En línea
 - ID
 - tiene que ser unico
 - sintaxis #nombre:id
 - CLASS
 - puede ser repetido
 - sintaxis .nombre_class
 - p.nombre_class
 - afectar a todas las p con esa clase
 - especifico
 - *{ }
 - todos los elementos
 - h1,h2, p
 - Agrupación
 - !important

Tema 5 - JS

- **JavaScript (JS)**
 - Creado originalmente para que las webs estuvieran vivas
 - Orientado a objetos
 - Version de servidor
 - node.js

Tema 6 - Node.js

- Open source
- Cross-platform
 - nvm
 - node version manager
 - trabajar con diferentes versiones de node.js
 - npm
 - node package manager
 - Gestor de paquetes de node.js
 - instalado junto con node.js
 - JSON
 - JavaScript Object Notation
 - Formato ligero para el intercambio de datos
 - Soporta objetos, arrays y valores
 - No permiten comentarios
 - Objeto
 - set desordenado de pares clave/valor
 - Rodeado por llaves { }
 - La clave es un string y tiene que tener comillas dobles
 - "clave" : "valor"
 - La clave debería usar nomenclatura lower camel case
 - datos se separan por comas
 - Array
 - colección ordenada de valores
 - rodeado por corchetes []

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
```

```
"city": "New York",
"state": "NY",
"postalCode": "10021-3100"
},
"phoneNumbers": [
  {
    "type": "home",
    "number": "212 555-1234"
  }, {
    "number": "646 555-4567"
  }
],
"children": [],
"spouse": null
}
```

- Testing
 - Buscar bugs
 - Evitar errores futuros
 - ejecución automática
- Continuous integration
 - Cada merge del código a la rama main implica
 - Code build
 - descarga de dependencias
 - Instalación de herramientas
 - compilación de código
 - linting errores de estilo
 - generación de la version final
 - test
 - unit test
 - integration test
 - end to end
 - Si el ci falla, se genera un informe
 - si el ci tiene éxito se publicara la version en producción

Tema 7 - Diseño y Accesibilidad

- Qué es la accesibilidad?
 - alcance todo tipo de personas
 - cualquier tipo de persona puede acceder a cualquier cosa
 - Accesible para todo el mundo
- Accesibilidad web
 - cualquier objeto sea utilizado por todo el público

- Proporcionar un texto alternativo
- Proporcionar encabezados
- Subtitula o proporciona transcripciones
- Permite saltar los elementos repetitivos
- Asegura que el texto es claro y fácil de leer
- Accesibilidad del contenido
- WCAG
 - Web Content Accessibility Guidelines
 - estándar para accesibilidad web
- ARIA
 - Accessible Rich Internet Applications
 - Serie de roles y atributos para hacer la web mas accesible