Sesión 10
Herramientas de modelado, PlantUML.

Objetivo

Dar a conocer a los participantes, los fundamentos y técnicas de uso de la herramienta de modelado “PlanUML” para diseñar bases de datos y sus componentes, a través de la correspondiente presentación y la demostración de un caso ejemplo, en una sesión expositiva-demostrativa.

Requerimientos

- PC o Equivalente con Windows o Linux.
- Navegador web (Firefox, Chrome, Brave, Opera, etc.)
- Gestor de base de datos MySql/MariaDB/PostgreSQL
- Cliente de base de datos

Requisitos

- Haber asistido a las sesiones 6, 7, 8 y 9 de la unidad didáctica.

Sesión 10
Herramientas de modelado, PlantUML.

¿Qué es PlantUML?

PlantUML es una herramienta de código abierto que permite crear diagramas a partir de lenguaje descriptivo en texto plano. Utiliza Graphviz como motor de renderizado y soporta múltiples tipos de diagramas mediante sintaxis específica.

Historia y Orígenes

•**Creado por:** Arnaud Roques
•**Año de creación:** 2009
•**Inspiración:** Necesidad de mantener diagramas en control de versiones
•**Evolución:** De diagramas UML simples a múltiples tipos de diagramas

Filosofía del Proyecto

"Los diagramas deben ser tan fáciles de mantener como el código"

- **Texto plano → Diagramas**
- **Control de versiones amigable**
- **Independiente de plataforma**

Arquitectura de PlantUML

Componentes Principales:

Texto PlantUML → PlantUML Processor → Graphviz → Imagen/PDF/SVG

Sesión 10
Herramientas de modelado, PlantUML.

Tecnologías Subyacentes:

- **Java:** Lenguaje base
- **Graphviz:** Motor de renderizado
- **DOT Language:** Formato de Graphviz
- **Múltiples salidas:** PNG, SVG, PDF, ASCII Art

Características Principales

1. Multiplataforma

```bash
# Funciona en cualquier SO con Java
java -jar plantuml.jar diagrama.txt
```

2. Múltiples Tipos de Diagramas

- Diagramas UML (Clases, Secuencia, Casos de Uso)
- Diagramas ER (Entidad-Relación)
- Diagramas de Arquitectura
- Mapas Mentales
- Wireframes

3. Control de Versiones

```git
# Los archivos .puml son texto plano
git add diagrama.puml
git commit -m "Actualiza diagrama ER"
```

Sesión 10
Herramientas de modelado, PlantUML.

4. Integración Extensa

- **IDEs:** VS Code, IntelliJ, Eclipse
- **Documentación:** Confluence, Jira, MkDocs
- **CI/CD:** Jenkins, GitHub Actions
- **Editores Online:** PlantUML Server

Sintaxis Básica Fundamental

1. Estructura Básica

plantuml

```
@startuml
' Comentarios con apóstrofe
!theme plain

' Contenido del diagrama aquí...

@enduml
```

2. Entidades y Atributos

plantuml

```
entity "Nombre_Entidad" as alias {
  * campo1 : tipo [PK]
  --
  campo2 : tipo
  campo3 : tipo [FK]
}
```

Sesión 10
Herramientas de modelado, PlantUML.

3. Relaciones y Cardinalidad

plantuml

```
' Sintaxis general:
Entidad1 [cardinalidad1]--[cardinalidad2] Entidad2 : "etiqueta"

' Cardinalidades comunes:
```

- ||--|| : Uno a Uno (mandatorio)
- }|--|| : Muchos a Uno
- }|--{| : Muchos a Muchos
- ||--o{ : Uno a Muchos (uno mandatorio)
- o|--o{ : Muchos a Muchos (opcionales)

4. Configuración Visual (Skinparams)

plantuml

```
skinparam linetype ortho
skinparam shadowing false
skinparam classFontSize 14
skinparam classFontStyle bold
skinparam classBackgroundColor LightBlue
```

Sesión 10
Herramientas de modelado, PlantUML.

Comandos Específicos Usados

Temas Predefinidos

plantuml

```
!theme plain
!theme blueprint
!theme dark
!theme cerulean
```

Ocultar Elementos

plantuml

```
hide circle        ' Oculta círculo en entidades
hide empty members ' Oculta secciones vacías
```

Estilos de Líneas

plantuml

```
skinparam linetype polyline  ' Líneas curvas
skinparam linetype ortho     ' Líneas rectas (usado)
```

Sesión 10
Herramientas de modelado, PlantUML.

Ventajas vs Desventajas

Ventajas:

- **Versionable:** Diferencias claras en Git
- **Escalable:** Fácil refactorizar
- **Automátizable:** Generación en build processes
- **Consistente:** Mismo estilo en todos los diagramas
- **Gratuito:** Open Source

Desventajas:

- **Curva de aprendizaje:** Sintaxis específica
- **Menos visual:** No WYSIWYG
- **Dependencias:** Requiere Java/Graphviz
- **Personalización limitada:** Comparado con herramientas visuales

Flujo de Trabajo Típico

Desarrollo Iterativo:

- Editar archivo .puml
- Generar diagrama
- Revisar resultado
- Ajustar sintaxis
- Commit cambios

Sesión 10
Herramientas de modelado, PlantUML.

Ejemplo de Pipeline:

```yaml
# GitHub Actions example
- name: Generate diagrams
  run: java -jar plantuml.jar docs/diagrams/
```

Extensiones y Herramientas Relacionadas

Ecosistema PlantUML:

- **C4-PlantUML:** Diagramas de arquitectura
- **PlantUML GitHub Action:** Automatización
- **PlantText:** Editor online
- **PlantUML QEditor:** Editor desktop
-

Sesión 10
Herramientas de modelado, PlantUML.

Bibliografía

- "Desarrollo de Aplicaciones Android con App Inventor 2", TOMAS DOMINGUEZ MINGUEZ,
  Ed. Marcombo, 2022.
- "Crea tus Aplicaciones Android con app Inventor 2", José Manuel Cabello García Ed. IC.
- "Android - Guía para desarrolladores", Ableson, F., Sen, R., King, C. Ed. Anaya Multimedia.
- "El Gran Libro de Android", Gironés, J.
  Ed. Alfaomega. Ediciones de la U, 2023;
  ISBN 9587924649, 9789587924640; N.º de páginas:316.
- https://docs.oracle.com/en/database/