

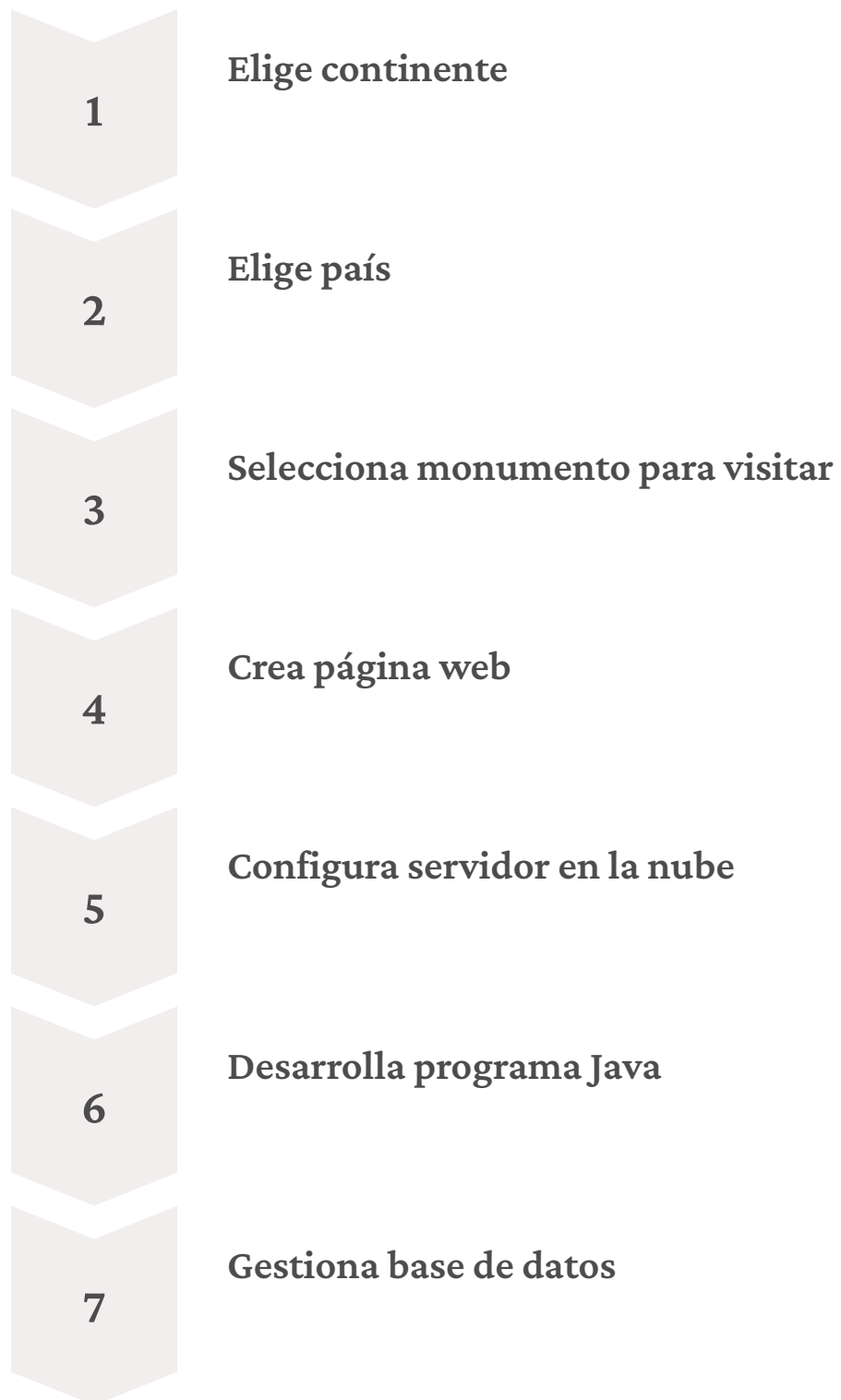
Proyecto Intermodular - Sistema de Gestión de Monumentos

Subtítulo:

- Base de datos relacional
- Aplicación Java
- Sitio web interactivo
- Despliegue en la nube

 por Pep AH

Flujo del Proyecto



Estructura de la Base de Datos

Tablas:

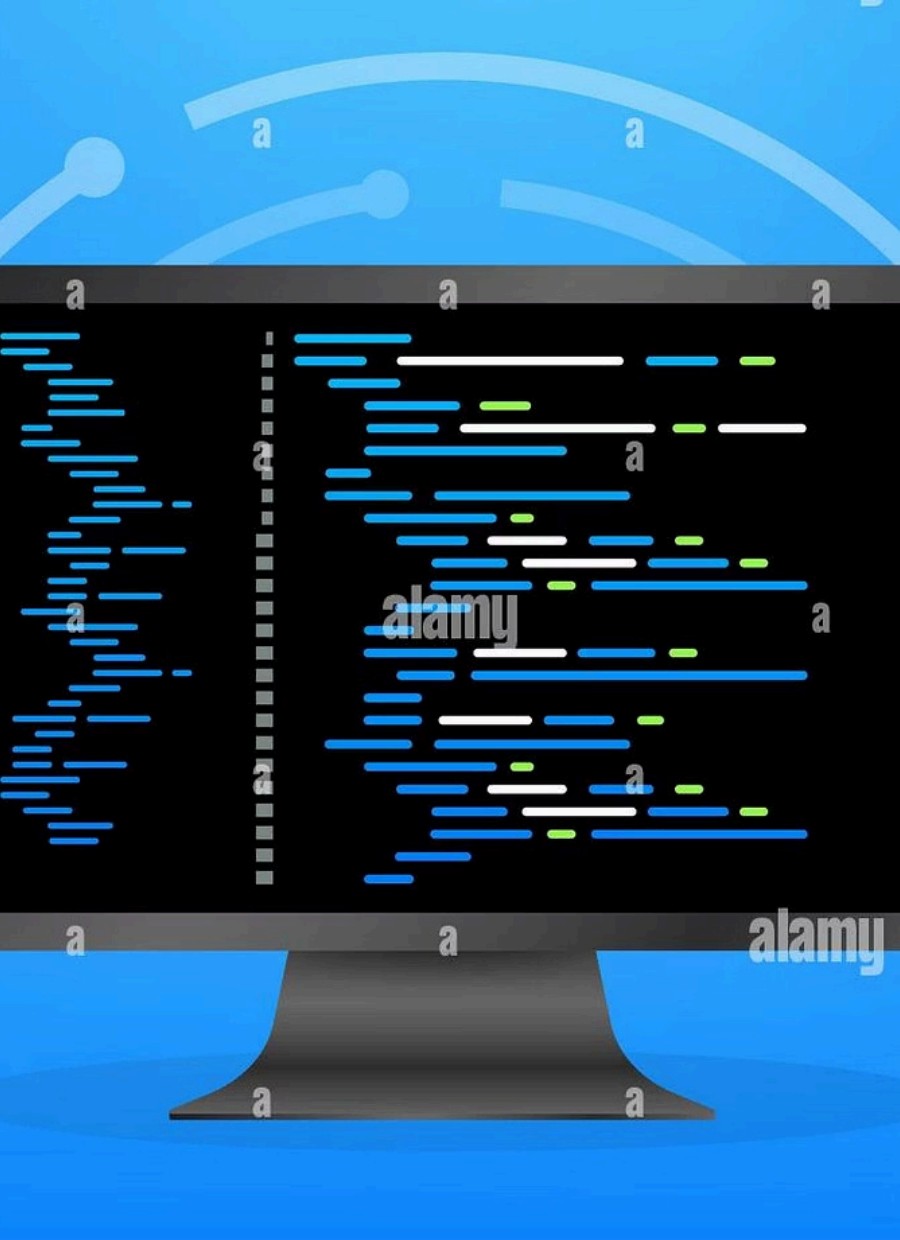
- **continentes** (id, nombre, superficie)
- **paises** (id, nombre, capital, FK continente)
- **monumentos** (id, nombre, tipo, FK país)

Relaciones:

1-N entre tablas

Triggers:

- Registro automático de fechas
- Conteo de monumentos por país



Conexión Java-MySQL

Clases:

- **Continentes, Países, Monumentos** (POJOs)
- **DataBaseManager** (gestión de conexiones y consultas)

Funcionalidades:

- Consulta de datos jerárquicos
- Manejo de excepciones SQL

preview.colorlib.com

Table #09

CUSTOMER	SHIP	PRICE	PRUCHASED PRICE	STATUS
Mark Otto	Japan	\$3000	\$1200	Progr
Mark Otto	Japan	\$3000	\$1200	Oper
Mark Otto	Japan	\$3000	\$1200	On h
Mark Otto	Japan	\$3000	\$1200	Progr
Mark Otto	Japan	\$3000	\$1200	On h
Mark Otto	Japan	\$3000	\$1200	Oper
Mark Otto	Japan	\$3000	\$1200	Oper
Mark Otto	Japan	\$3000	\$1200	Progr

Página Web Estática

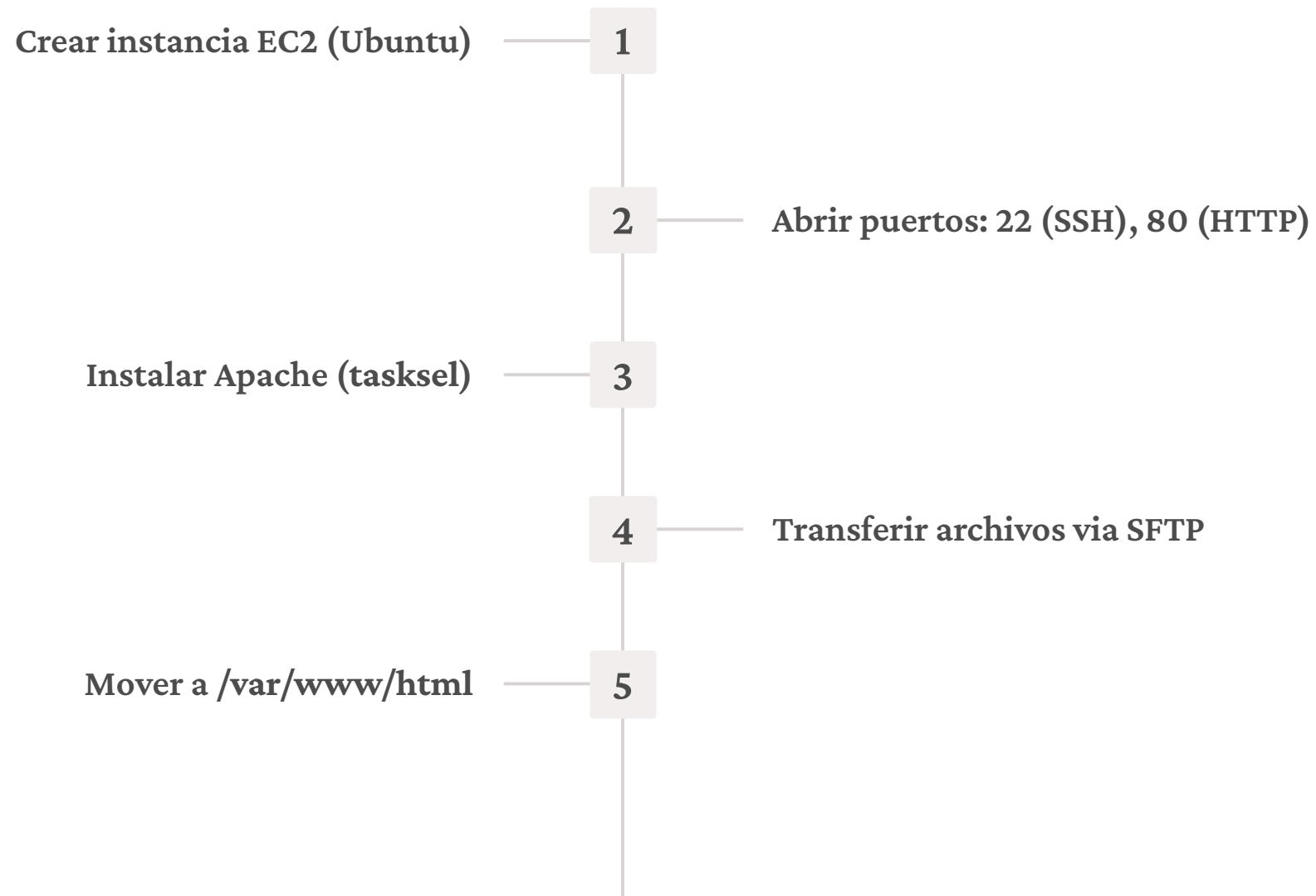
HTML con tablas de:

- Continentes (nombre, superficie)
- Países (nombre, capital, población)
- Monumentos (nombre, tipo, patrimonio UNESCO)

Estructura semántica

(thead, tbody)

Infraestructura en AWS



Requisitos:

- Clave .ppk para autenticación
- Permisos sudo para despliegue

Resultados Visuales



Tablas de
datos en
navegador



Consola de
AWS
(instancia
activa)



Terminal con
comandos de
instalación



Página web
accesible
públicamente

Capturas de:



Logros y Aprendizaje

Integración exitosa de:

- Backend (MySQL + Java)
- Frontend (HTML estático)
- DevOps (AWS EC2 + Apache)

Habilidades adquiridas:

- Modelado de bases de datos
- Programación orientada a objetos
- Administración de servidores

