¡Entendido! Aquí tienes el contenido detallado para cada módulo del curso de administración avanzada de MediaWiki:

**Módulo 1: Configuración Avanzada de MediaWiki**

**Duración:** 1 hora

**Contenido Específico:**

**1. Repaso rápido de la instalación básica**

* Comando para instalar MediaWiki en un servidor local:
* php maintenance/install.php --dbname=miwiki --dbuser=root --pass=mi\_password --server="http://localhost" --scriptpath="" MiWiki Admin
* Archivos clave: LocalSettings.php.

**2. Configuración avanzada de LocalSettings.php**

* Modificación de idioma:
* $wgLanguageCode = "es";
* Habilitación de depuración:
* $wgDebugLogFile = "/var/log/mediawiki/debug.log";
* Configuración de zona horaria:
* $wgLocaltimezone = "America/Mexico\_City";
* $wgLocalTZoffset = date("Z") / 3600;

**3. Configuración de la caché:**

* Usando Memcached:
* $wgMainCacheType = CACHE\_MEMCACHED;
* $wgMemCachedServers = ["127.0.0.1:11211"];

**Ejercicio Práctico:**

1. Configura MediaWiki para que use el idioma español y habilita la depuración.
2. Simula la configuración de Memcached (asegúrate de que MediaWiki no arroje errores).

**Módulo 2: Gestión de Usuarios y Permisos**

**Duración:** 1 hora

**Contenido Específico:**

**1. Creación de grupos personalizados:**

$wgGroupPermissions['SuperModeradores']['edit'] = true;

$wgGroupPermissions['SuperModeradores']['delete'] = true;

**2. Configuración de permisos avanzados:**

* Asignar usuarios a grupos:
* $wgAddGroups['sysop'] = ['SuperModeradores'];
* $wgRemoveGroups['sysop'] = ['SuperModeradores'];

**3. Uso de LDAP Authentication:**

* Instalación:
* composer require mediawiki/ldap-authentication
* Configuración básica en LocalSettings.php:
* $wgLDAPDomainNames = ["example.com"];
* $wgLDAPServerNames = ["example.com" => "ldap.example.com"];

**Ejercicio Práctico:**

1. Crea un grupo llamado Colaboradores con permisos limitados a edición.
2. Configura un grupo llamado SuperModeradores y habilita la integración con un servidor LDAP ficticio.

**Módulo 3: Extensiones Avanzadas**

**Duración:** 1.5 horas

**Contenido Específico:**

**1. Instalación y configuración de extensiones:**

* Instalación de VisualEditor:
* git clone https://gerrit.wikimedia.org/r/mediawiki/extensions/VisualEditor extensions/VisualEditor

Configuración:

wfLoadExtension('VisualEditor');

**2. Configuración de Semantic MediaWiki:**

composer require mediawiki/semantic-mediawiki

wfLoadExtension('SemanticMediaWiki');

enableSemantics('example.com');

**3. Creación de una extensión básica:**

* Archivo: extensions/MiExtension/MiExtension.php:
* <?php
* if (function\_exists('wfLoadExtension')) {
* wfLoadExtension('MiExtension');
* return;
* }

**Ejercicio Práctico:**

1. Configura y activa **Semantic MediaWiki**.
2. Crea una etiqueta mágica personalizada MiEtiqueta.

**Módulo 4: Personalización del Tema y Skins**

**Duración:** 1 hora

**Contenido Específico:**

**1. Modificación de un skin existente:**

* Modificar Vector para cambiar el logo:
* $wgLogo = "$wgResourceBasePath/resources/assets/mi-logo.png";

**2. Creación de un skin personalizado:**

* Crear un archivo en skins/MiSkin/MiSkin.php:
* <?php
* $wgValidSkinNames['miSkin'] = 'MiSkin';

**Ejercicio Práctico:**

1. Cambia los colores principales de un skin.
2. Añade un pie de página único al skin Vector.

**Módulo 5: Optimización de Rendimiento**

**Duración:** 1 hora

**Contenido Específico:**

**1. Configuración de caché avanzada:**

* Configuración del caché de páginas:
* $wgCachePages = true;

**2. Optimización de consultas SQL:**

* Habilitar $wgQueryCacheLimit.

**Ejercicio Práctico:**

1. Activa el caché de páginas y evalúa el rendimiento.

**Módulo 6: Integraciones Externas y APIs**

**Duración:** 1 hora

**Contenido Específico:**

**1. Uso de la API de MediaWiki:**

* Ejemplo básico para obtener una página:
* curl "http://example.com/api.php?action=query&titles=Main\_Page&format=json"

**2. Creación de un bot con Python:**

* Script básico:
* import requests
* API\_URL = "http://example.com/api.php"
* data = {
* "action": "edit",
* "title": "Página de Prueba",
* "text": "Contenido generado automáticamente",
* "token": "TOKEN"
* }
* response = requests.post(API\_URL, data=data)

**Ejercicio Práctico:**

1. Crea un script en Python que edite un artículo automáticamente.

**Módulo 7: Seguridad y Respaldo**

**Duración:** 1 hora

**Contenido Específico:**

**1. Configuración de HTTPS:**

* Configuración en Apache:
* sudo a2enmod ssl
* sudo service apache2 restart

**2. Creación de respaldos automáticos:**

* Script de respaldo:
* mysqldump -u root -p miwiki > backup.sql

**Ejercicio Práctico:**

1. Configura un cron job para respaldos automáticos.

**Módulo 8: Proyecto Final**

**Duración:** 1.5 horas

**Contenido Específico:**

1. Crear un entorno funcional que incluya:
   * Un tema personalizado.
   * Extensiones avanzadas configuradas.
   * Sistema de autenticación externo.

**Ejercicio Práctico:**

1. Implementa un sistema de documentación interno con MediaWiki.