
PROYECTO

201602404 - Kevin Estuardo Secaida Molina
201900221 - Gabriel Enrique Perez Meza
202002364 - Dayana Alejandra Reyes Rodríguez
202010816 - César André Ramírez Dávila

Resumen

Se realizó un programa capaz de presentar información de forma más cómoda tanto para los clientes como para los trabajadores de la empresa Desktop Records.

El sistema tiene la capacidad de recibir información de sus clientes, discos musicales y los países donde la plataforma tiene alcance.

Esta información se recibe en formato XML (eXtensible Markup Language), el sistema decodifica e ingresa la información a su base de datos para poder manipularla y presentar dicha información de manera eficiente.

Ya con la información, el sistema realiza reportes gráficos para visualizar la información.

Abstract

A program capable of presenting information in a more comfortable way for both clients and workers of the Desktop Records company was created.

The system can receive information from its clients, music records and the countries where the platform has reach.

This information is received in XML format (eXtensible Markup Language), the system decodes and enters the information into its database to be able to manipulate it and present said information efficiently.

Already with the information, the system makes graphic reports to visualize the information.

Palabras clave

- XML
- Programa
- Sistema
- Información
- Reporte

Keywords

- XML
- Program
- System
- Information
- Report

Introducción

La empresa Desktop Records, decide que es momento de presentar su información de una manera más cómoda para los usuarios, por lo que se desea implementar un sitio web capaz de procesar, manipular y presentar la información de la empresa.

Se realizó un sitio web que tiene la capacidad de recibir una gran cantidad de información, esta debe tener un formato XML.

El sitio web se compone de dos elementos:

- ✓ **Backend:** Es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos como, por ejemplo, la comunicación con el servidor.
- ✓ **Frontend:** Es parte de un sistema de información al que el usuario accede directamente e interactúa para recibir o utilizar las capacidades de backend del sistema anfitrión. Permite a los usuarios acceder y solicitar las prestaciones y servicios del sistema de información subyacente. El sistema de frontend puede ser una aplicación de software o hardware o su combinación, así como recursos de la red

Desarrollo del tema

Los mensajes HTTP, son los medios por los cuales se intercambian datos entre servidores y clientes. Hay dos tipos de mensajes: peticiones, enviadas por el cliente al servidor, para pedir el inicio de una acción; y respuestas, que son la respuesta del servidor.

Peticiones usadas:

GET: El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

- **GET - /empleados:** Devolverá la información de todos los empleados de cada departamento.
- **GET - /reporteEmpleados:** Esta petición ejecutará código para generar el gráfico de Graphviz con la información de los empleados.
- **GET - /discos:** Devolverá la información de todos los discos.
- **GET - /reporteDiscos:** Esta petición ejecutará código para generar el gráfico de Graphviz con la información de los discos del catálogo.
- **GET - /países:** Devolverá la información de todos los países de cada continente.
- **GET - /reporteRegiones:** Esta petición ejecutará código para generar el gráfico de Graphviz con la información de los países registrados en la plataforma.

POST: El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

- **POST - /empleadoNombre:** Se enviará un nombre y la petición devolverá la información de los empleados que contengan el nombre indicado.
- **POST - /empleadoDepartamento:** Se enviará un departamento y la petición devolverá la información de los empleados que pertenezcan al departamento indicado.
- **POST - /empleadoSueldo:** Se enviará un sueldo y la petición devolverá la información de los empleados que contengan el sueldo indicado
- **POST - /agregarEmpleado:** Se enviará una estructura JSON con la información del empleado (departamento, ID, nombre, puesto, salario) y la petición ejecutará código para poder ingresar el empleado al departamento. No se evaluarán los casos de errores ID repetido, o departamento inexistente, por lo que no es necesario realizar esas validaciones.

- **POST - /modificarEmpleado** Se enviará una estructura JSON con la información del empleado (ID, nombre, puesto, salario) y la petición ejecutará código para poder modificar el empleado con el ID enviado.
- **POST - /eliminarEmpleado:** Se enviará una estructura JSON con la información del empleado (ID) y la petición ejecutará código para poder eliminar el empleado del departamento.
- **POST - /discoTitulo:** Se enviará un nombre y la petición devolverá la información de los discos que contengan el nombre indicado.
- **POST - /discoYear:** Se enviará un año y la petición devolverá la información de todos los discos que pertenezcan al año indicado.
- **POST - /discoArtista:** Se enviará un nombre y la petición devolverá la información de todos los discos que contengan el nombre indicado.
- **POST - /agregarDisco:** Se enviará una estructura JSON con la información del disco (título, artista, país, compañía, precio y año) y la petición ejecutará código para poder ingresar el disco al catálogo.
- **POST - /modificarDisco:** Se enviará una estructura JSON con la información del disco (título, artista, país, compañía, precio y año) y la petición ejecutará código para poder modificar el disco con el título indicado.
- **POST - /eliminarDisco:** Se enviará una estructura JSON con la información del disco (título) y la petición ejecutará código para poder eliminar el disco del departamento.
- **POST - /paisMoneda:** Se enviará un tipo de moneda y la petición devolverá la información de los países que contengan la moneda indicada.
- **POST - /paisIdioma:** Se enviará un idioma y la petición devolverá la información de todos los países que contengan el idioma indicado
- **POST - /continente:** Se enviará un nombre de continente y la petición devolverá la información de todos los países de ese continente.

- **POST - /agregarPais:** Se enviará una estructura JSON con la información del país (continente, nombre, moneda, capital, idioma, población) y la petición ejecutará código para poder ingresarlo.
- **POST - /modificarPais:** Se enviará una estructura JSON con la información del país (continente, nombre, moneda, capital, idioma, población) y la petición ejecutará código para poder modificarlo.

Se utilizaron herramientas para el desarrollo de las aplicaciones como frameworks o librerías.

Un framework es un marco de trabajo y conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Para el desarrollo de la plataforma en la parte del frontend se utilizó el framework Django.

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC). Fue desarrollado originalmente para gestionar páginas web orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt.



Figura 1. Logo Django.

Fuente: Página oficial de django
<https://www.djangoproject.com/>

Para la parte del backend se hizo uso del lenguaje de programación Python; se utilizaron los servicios del framework Flask para crear un servidor local.

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Spotify, Panda 3D, entre otros. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.



Figura 2. Logo Python.

Fuente: Página oficial de python
<https://www.python.org/>

Flask es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.



Figura 3. Logo Flask.

Fuente: Página oficial de flask
<https://flask.palletsprojects.com/en/2.1.x/>

Se hizo uso del versionamiento para poder realizar esta plataforma, alojando así nuestro proyecto en la nube con la ayuda de Github y un repositorio en esta plataforma.

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena generalmente de forma pública.



Figura 4. Logo Github.

Fuente: Página pngimg
<https://pngimg.com/image/73372>

Para escribir el programa se utilizó un editor de código fuente, VS Code.

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto,¹² aunque la descarga oficial está bajo software

privativo e incluye características personalizadas por Microsoft.



Figura 5. Logo VS Code.

Fuente: Página oficial de VS Code
<https://code.visualstudio.com/>

Referencias bibliográficas

Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. Ciencias Holguín, 20(2), 1-13.

Forcier, J., Bissex, P., & Chun, W. J. (2008). Python web development with Django. Addison-Wesley Professional.

Grinberg, M. (2018). Flask web development: developing web applications with python. " O'Reilly Media, Inc."

Marzal Varó, A., & Gracia Luengo, I. (2009). Introducción a la programación con Python. Universitat Jaume I.

Conclusiones

- Los mensajes HTTP son la clave para usar HTTP; su estructura es sencilla y son fácilmente ampliables.
- Es necesario hacer uso del versionamiento para poder desarrollar de la manera más eficaz un proyecto.
- Los frameworks son herramientas que facilitan al desarrollador.