
PROYECTO 3 SEGUNDO SEMESTR2 2022

202106484 – Wendi Paulina Vicente Pérez

Resumen

El protocolo de transferencia de hipertexto seguro, más conocido como HTTPS, es la versión segura de HTTP. HTTP es el principal protocolo utilizado para enviar datos entre un navegador y un sitio web. A su vez, HTTPS usa el cifrado para aumentar la seguridad en la transferencia de datos.

Esto es crucial cuando los usuarios transmiten datos sensibles, como al iniciar sesión en una aplicación bancaria, un servicio de correo electrónico o cualquier otro servicio similar.

HTTPS tiene un componente de cifrado de datos que se conoce como TLS/SSL, y que utiliza certificados del estándar X.509. Este es un estándar de clave pública que se utiliza para crear certificados digitales capaces de permitir a los usuarios autenticar a los remitentes de los mensajes. Con ellos, los navegadores web se aseguran de que un servidor pertenece realmente al dominio al que se quiere acceder.

Además, este protocolo seguro utiliza el puerto 443 y no el 80, el habitual para HTTP. Los puertos forman parte de una dirección de red y permiten una asignación exacta de las conexiones entre el cliente y el servidor.

Palabras clave

Backend
Frontend
Archivo xml

Abstract

Hypertext Transfer Protocol Secure, better known as HTTPS, is the secure version of HTTP. HTTP is the main protocol used to send data between a browser and a website. In turn, HTTPS uses encryption to increase the security of data transfer.

This is crucial when users transmit sensitive data, such as when logging into a banking application, email service, or any other similar service.

HTTPS has a data encryption component known as TLS/SSL, which uses certificates from the X.509 standard. This is a public key standard used to create digital certificates capable of allowing users to authenticate the senders of messages. With them, web browsers make sure that a server really belongs to the domain they want to access.

In addition, this secure protocol uses port 443 and not 80, the usual one for HTTP. Ports are part of a network address

and allow precise connections between client and server.

Keywords

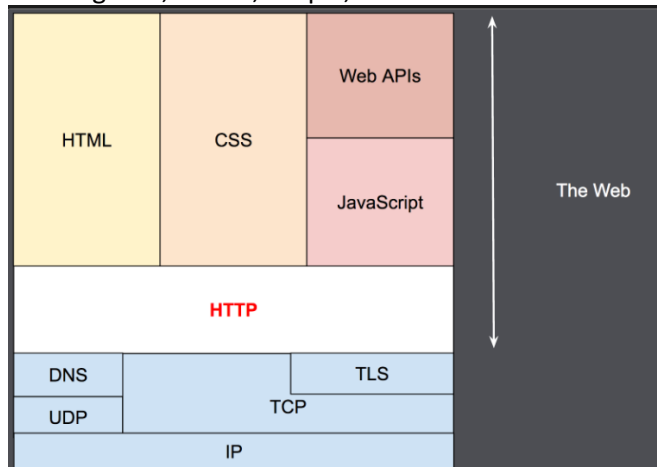
Backend
Frontend
xml file

Introducción

Muchos de los programas se trabajan por medio de backend y frontend ya que es lo mas solicitado entre los usuarios de muchas empresas por su facilidad de manejo y seguridad de los datos en servidores. Es por eso que se trabajo en un programa donde se obtienen datos de un XML y poder facturar de una manera mas fácil.

Desarrollo del tema PROTOCOLO HTTP

HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Así, una página web completa resulta de la unión de distintos sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc...



Backend

El 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.

El encargado de desarrollar el back-end se pasa el día entero toqueteando diferentes y variados códigos de programación en lenguajes como PHP, Java o .Net. El desarrollador debe ser meticuloso y cuidadoso en su trabajo ya que un pequeño despiste y ¡bum!... Página web caída.

Frontend

El frontend sirve para realizar la interfaz de un sitio web, desde su estructura hasta los estilos, como pueden ser la definición de los colores, texturas, tipografías, secciones, entre otros. Su uso es determinante para que el usuario tenga una buena experiencia dentro del sitio o aplicación.

Conclusión

En base a la información planteada y las necesidades de nuestro proyecto se puede decir que la implementación de https para la gestión de nuestro proyecto es una muy buena opción ya que cuentan con métodos que nos facilitan la gestión de información que se manejara de cada uno de nuestras empresas, puntos de atención, escritorios y puntos de atención , también hay que tomar en cuenta que se puede implementar listas enlazada independientemente del tipo que esta sea como lo es la lista simplemente enlazada o doblemente enlazada esto dependerá de la facilidad que estas nos proporcionan en base a lo que se desea trabajar con ellas al igual que siempre tener en cuenta un buen manejo de versiones en cualquier herramienta que nos ayude con este manejo

Referencias

Copyright © 2006 Guillem Borrell Nogueras El control de versiones

<https://sites.google.com/site/estdatinfjiq/unidad-iii-listas-enlazada>

Anexos:

OBJETIVO GENERAL

Desarrollar una solución integral que implemente un API que brinde servicios utilizando el Protocolo HTTP bajo el concepto de programación orientada a objetos (POO) y el uso de bases de datos.

OBJETIVOS ESPECÍFICOS

- Implementar un API a través de lenguaje Python que pueda ser consumida utilizando el protocolo HTTP.
- Utilizar el paradigma de programación orientada a objetos para construir software.
- Utilizar bases de datos para almacenar información de forma persistente.
- Utilizar archivos XML como insumos para la comunicación con el API desarrollado.
- Utilizar expresiones regulares para extraer contenido de texto.

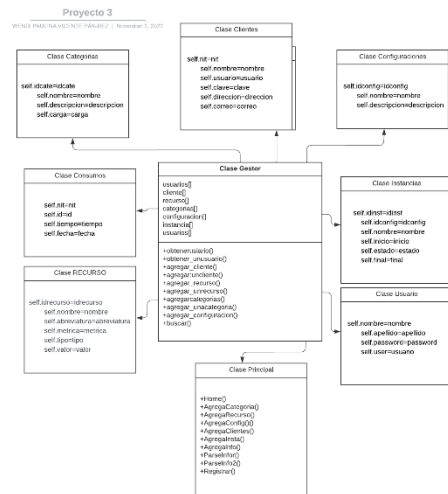
Mensaje de entrada

```
<?xml version="1.0"?>
<archivoConfiguraciones>
  <listaRecursos>
    <recurso id="idRecurso">
      <nombre> $nombreRecurso </nombre>
      <abreviatura> $abreviaturaRecurso </abreviatura>
      <metrica> $NombreMetrica </metrica>
      <tipo> $tipoRecurso </tipo>
      <valorXhora> valorNumerico </valorXhora>
    </recurso>
    ...
  </listaRecursos>
  <listaCategorias>
    <categoria id="idCateoria">
      <nombre> $nombreCategoria </nombre>
      <descripcion> $descripcionCategoria </descripcion>
      <cargaTrabajo> $descripcionCargaTrabajo </cargaTrabajo>
      <listaConfiguraciones>
        <configuracion id="idConfiguracion">
          <nombre> $nombreConfiguracion </nombre>
          <descripcion> $descripcionConfiguracion </descripcion>
          <recursosConfiguracion>
            <recurso id="idRecurso"> cantidadRecurso </recurso>
          </recursosConfiguracion>
        </configuracion>
        ...
      </listaConfiguraciones>
    </categoria>
    ...
  </listaCategorias>
  <listaClientes>
    <cliente id="idCliente">
      <nombre> $nombreCliente </nombre>
      <descripcion> $descripcionCliente </descripcion>
      <estado> $estado </estado>
    </cliente>
    ...
  </listaClientes>
</archivoConfiguraciones>
```

Mensaje para notificar consumo

```
<?xml version="1.0"?>
<listadoConsumos>
  <consumo nitCliente="$nit" idInstancia="$id">
    <tiempo> tiempoConsumido </tiempo>
    <fechaHora> $descripcionFechaHora </fechaHora>
  </consumo>
  ..
</listadoConsumos>
```

Diagrama de Clases:



Arquitectura ARQUITECTURA

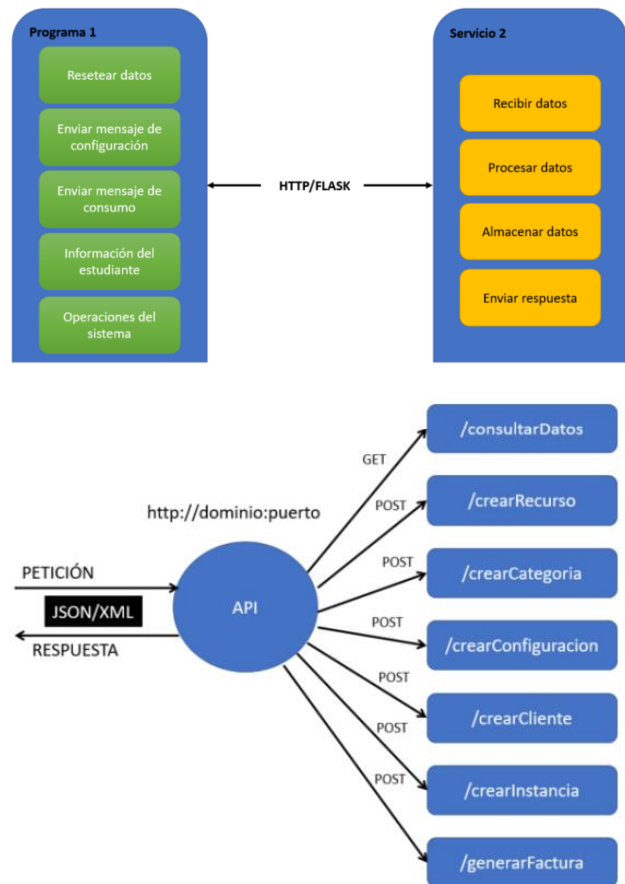


Fig. 4 - Ejemplo de la estructura de un API

