BIBLIOTECA FESC

PRESENTADO A:

ALBERT PAEZ

REALIZADO POR:

HUGO CAÑAS

ESNEYDER QUIÑONES

ERICK OCAMPO

FUNDACION ESTUDIOS COMFANORTE (FESC)

CUCUTA NDS

2023

PROYECTO BIBLIOTECA FESC

RESUMEN

Este proyecto describe la creación de una página web, la cual va a ser empleada para poder realizar, prestar libros, sancionar por no devolver libros para esto se usará un usuario y dar de baja a un libro. Para este proyecto emplearemos la metodología scrum. Se realizará en varios “Sprints” de los que obtendremos los prototipos funcionales para poder avanzar en el proyecto y terminarlo en el tiempo establecido sin errores y completo, esperando poder realizar futuras actualizaciones cortas.

Emplearemos para realizar este proyecto: js y HTML.

まとめ

このプロジェクトでは、本の貸し出し、本を返却しない場合の制裁を実行するために使用される Web ページの作成について説明します。このプロジェクトでは、スクラム手法を使用します。それはいくつかの「スプリント」で実行され、そこから機能プロトタイプを取得してプロジェクトを進め、確立された時間内にエラーなしで完了して完了することができ、将来の短い更新を行うことができることを期待しています

このプロジェクトを実行するために使用します: js と HTML



AGRADECIMIENTOS

Me gustaría agradecer a la universidad FESC por brindarnos la oportunidad de participar en este proyecto tan importante como lo es la biblioteca de nuestra institución.

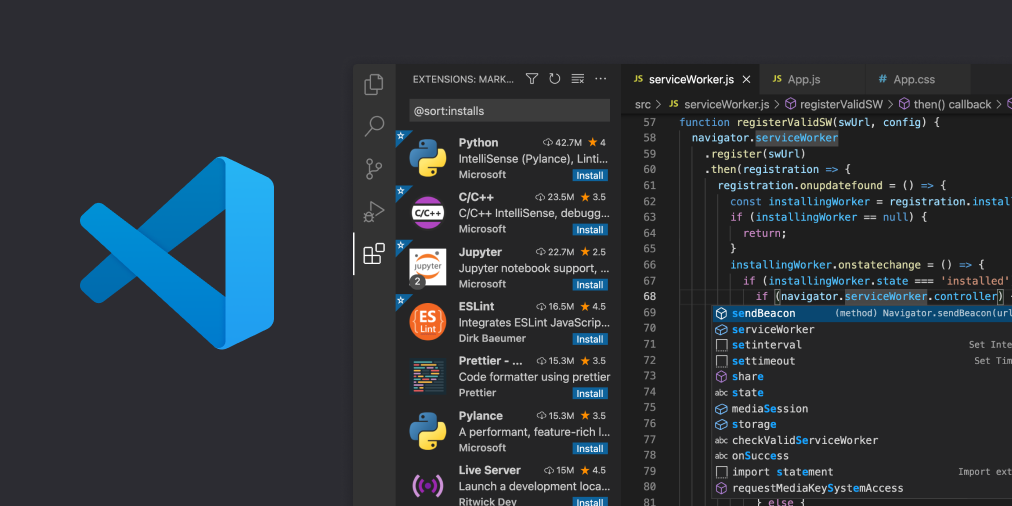
Agradezco a la perfecta sincronía y colaboración de mi equipo asignado ya que sin ellos este proyecto no podría haberse llevado acabo.

SISTEMA OPERATIVO



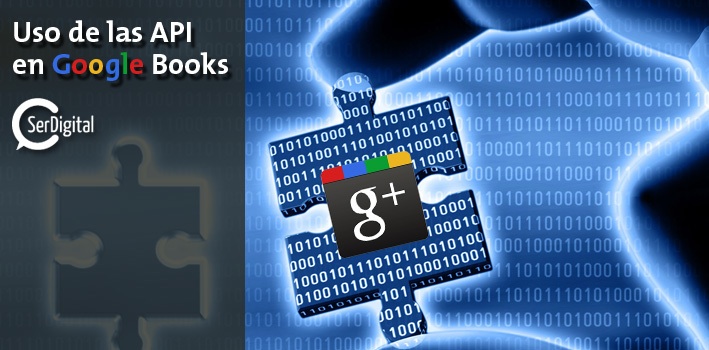
Windows 10 es un sistema operativo de la familia Windows NT desarrollado por Microsoft. Fue lanzado en 2015 y se ofreció como descarga gratuita durante un año a los usuarios que tuvieran copias genuinas de Windows 7 o Windows 8.1. Windows 10 introdujo una arquitectura de aplicaciones universales, una interfaz de usuario actualizada, la Vista de Tareas, Microsoft Edge y otras aplicaciones nuevas o actualizadas. La mayoría de las críticas positivas elogiaron la eliminación de la interfaz de usuario introducida por Windows 8 en un entorno no táctil para proporcionar una interfaz más orientada al escritorio. Sin embargo, los críticos criticaron la falta de control del usuario en algunas operaciones y las preocupaciones sobre la privacidad.

VISUAL STUDIO CODE



Visual Studio Code es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft para Windows, Linux, macOS y Web. Tiene varias características útiles, como soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Además, es personalizable, lo que permite a los usuarios cambiar el tema del editor, los atajos de teclado y las preferencias según sus necesidades. El software se basa en Electrón, un framework utilizado para implementar Chromium y Node.js como aplicaciones de escritorio, que se ejecuta en el motor de diseño Blink. A pesar de usar Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services. Cabe destacar que, aunque el software es de código abierto, la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

API



Si por definición un API nos permite consumir contenidos, el API de Google Books nos permite acceder a la base de datos de libros desde aplicaciones ajenas a Google y la forma en la que se accede es haciendo búsquedas de libros directamente en nuestro sitio u aplicación o teniendo vistas previas de libros, esto con la finalidad de que la información que mostramos permita al usuario ser redirigido a Google Books, al ser un API que desarrolla Google hay que cumplir con los términos de uso.

En el proceso de dar de alta un libro se hace la búsqueda del ISBN directamente en la base de datos de Google Books y desde nuestro sitio, obteniendo datos como:

Título

Autor.

Editorial.

Portada.

ISBN 10 y 13.

Año de Publicación.

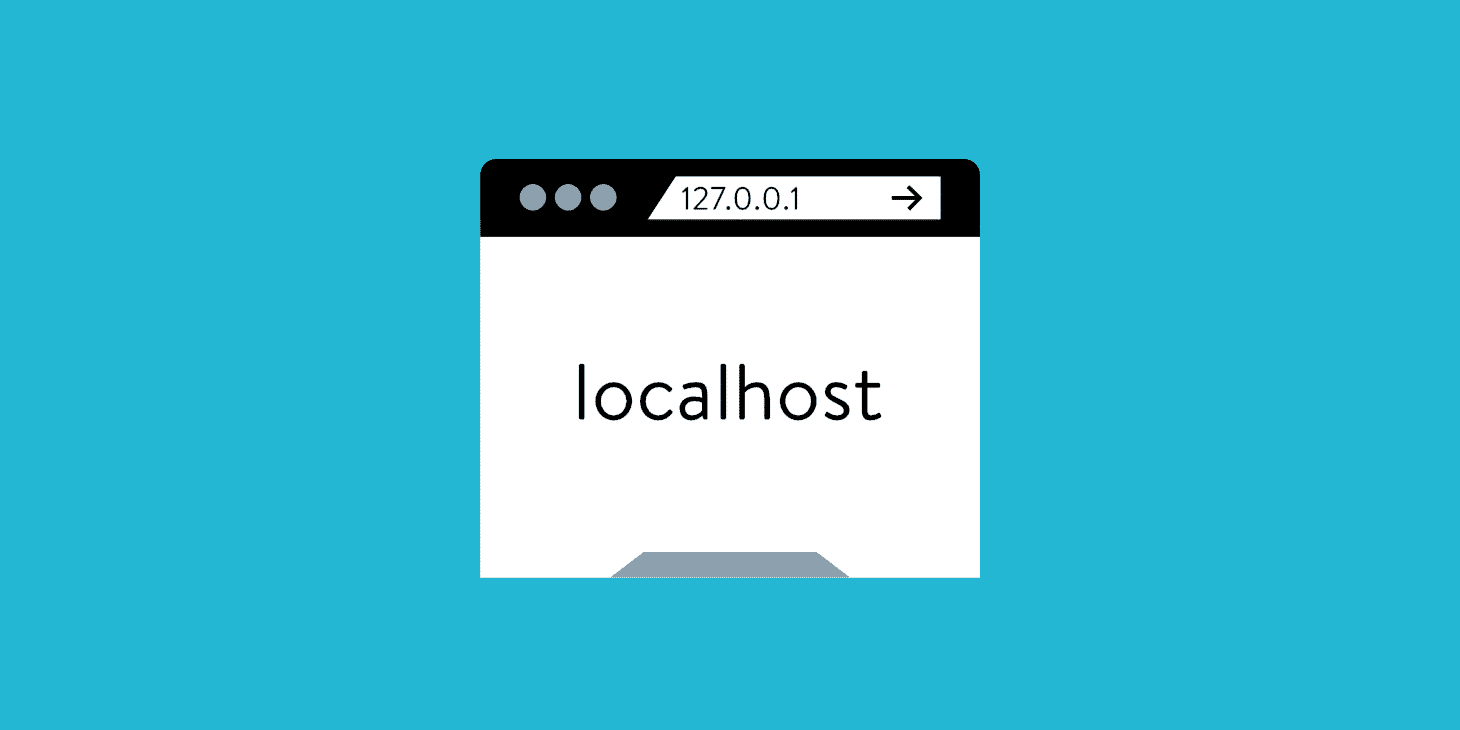
Portada (de estar disponible).

Eventualmente no se encontrará la información del libro, lo que implica ingresarlos de manera manual, el hacer nuestra implementación del API obedece a optimizar el tiempo en el momento de la captura de los datos bibliográficos del libro, limitándonos solo a verificar que la información que se obtiene de Google Books concuerde con la del libro.

VENTAJAS

Primeramente, tenemos acceso a una gran cantidad de información de libros y otra que nos permite hacer implementaciones personalizadas que consuman dicha información; en mi caso se está desarrollando un sistema de control de biblioteca.

LOCALHOST



Para definir el término «localhost» se comienza explicando que se refiere al propio ordenador que se tiene enfrente. Traducido literalmente al castellano, localhost es “huésped local”, pero en la práctica se utiliza para referirse al dispositivo o al servidor local. Así, cuando se llama al host local, el equipo habla, por así decirlo, consigo mismo o con un servidor instalado en él. No obstante, esta definición se queda corta, pues con localhost no se hace referencia de forma directa al propio PC, sino a su dirección IP. Cada ordenador suele tener una dirección IP que le identifica dentro de la red y otra con la cual el usuario navega en Internet, casi siempre dinámica y por normal general asignada por el proveedor de Internet. La dirección asociada al localhost, también llamada dirección IP de loopback, es especial e invariable y, si se utiliza un servidor en el propio equipo, sirve para acceder a él sin necesidad de salir a Internet.

XAMPP



Como te mencionamos antes, Xampp es un servidor web local multiplataforma que permite la creación y prueba de páginas web u otros elementos de programación. Sin embargo, Xampp integra una serie de herramientas que potencian y facilitan la experiencia al desarrollador. Es decir, Xampp en sí mismo no es un programa, sino un paquete de programas o software que contiene herramientas de gestión de base de datos. Fue desarrollado por Apache Friends y su nombre es un acrónimo de cada una de las principales herramientas que contiene; esto nos sirve para hacernos una idea de cuáles son los principales programas que vienen en el paquete.

X: Hace referencia a los diferentes sistemas operativos en los que se puede instalar el programa, como Linux, Windows, Mac, Ubuntu, etc.

A: Por el servidor web Apache que permitirá trabajar con las diferentes herramientas del paquete y que será la principal interfaz a usar. Este servidor web de código abierto será fácil de instalar gracias a Xampp.

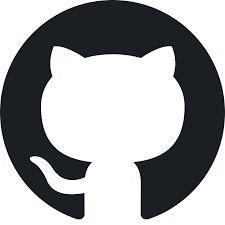
M: La letra M hace referencia a la incorporación del sistema de gestión de bases de datos conocido como MySQL; en algunas versiones de Xampp, sobre todo en las más recientes, incorpora MariaDB.

P: Como todo servidor web de código abierto, Xampp utiliza un lenguaje de programación, siendo en este caso PHP; este lenguaje es bastante conocido y soporta diferentes sistemas de bases de datos.

P: Perl es otro lenguaje de programación de Xampp, pero esta vez enfocado en la administración del sistema y programación de red.

Xampp incluye muchos otros componentes como PHPmyAdmin, o clientes FTP como Filezilla, durante la instalación del paquete podrás decidir si instalas o no.

GITHUB



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.

El 4 de junio de 2018 Microsoft compró GitHub por la cantidad de 7500 millones de dólares.1​2​ Al inicio, el cambio de propietario generó preocupaciones y la salida de algunos proyectos de este sitio;3​ sin embargo, no fueron representativos. GitHub continúa siendo la plataforma más importante de colaboración para proyectos de código abierto.

Link del repositorio donde se encuentra todo el proyecto: <https://github.com/HugoOrielso/Biblioteca-Fesc>

DEFINICION DE HISTORIA DE USUARIO

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| ID | ID: Identificador único asignado a este elemento del proyecto, con formato HUxx (siendo xx numeración progresiva de 01 a 99). |
| NOMBRE | • Nombre: Nombre corto utilizado para describir muy brevemente la historia de usuario. |
| PRIORIDAD | • Prioridad: Preferencia de cara al desarrollo de la historia de usuario respecto a las demás. Toma valores: Alta, media y baja. |
| RIESGO | • Riesgo: Se trata de la importancia de la historia de usuario en relación al conjunto del proyecto. Cuantificando de este modo el daño provocado en caso de fallo. Toma valores: Alto, medio y bajo. |
| DESCRIPCION | • Descripción: Breve explicación de las intenciones de la historia de usuario. Debe dejar clara la idea de la propia historia. |
| VALIDACION | • Validación: Condiciones que deben cumplirse una vez la historia está completamente desarrollada para que se pueda dar por finalizada. |

HISTORIAS DE USUARIOS DEL PROYECTO

|  |  |
| --- | --- |
| HISTORIA DE USUARIO #1 | |
| ID | HU01 |
| NOMBRE | Informes de usuarios 'deudores' |
| PRIORIDAD | Alta |
| RIESGO | Alta |
| DESCRIPCION | Como administradora de la biblioteca debo saber quién esta con un libro no entregado |
| VALIDACION | Quiero poder ver todos los deudores de libros |

|  |  |
| --- | --- |
| HISTORIA DE USUARIO #2 | |
| ID | HU02 |
| NOMBRE | Informe de artículos de la biblioteca |
| PRIORIDAD | Medio |
| RIESGO | Bajo |
| DESCRIPCION | Como administradora de la biblioteca debo saber el stock de libros |
| VALIDACION | Quiero poder conocer cuál es la totalidad de los libros |

|  |  |
| --- | --- |
| HISTORIA DE USUARIO #3 | |
| ID | HU03 |
| NOMBRE | Informe de usuarios |
| PRIORIDAD | Alta |
| RIESGO | Alta |
| DESCRIPCION | Como administradora debo guardar las personas que piden los libros |
| VALIDACION | Quiero poder archivar las personas que lleven un libro |

|  |  |
| --- | --- |
| HISTORIA DE USUARIO #4 | |
| ID | HU04 |
| NOMBRE | Informes acotados por fechas |
| PRIORIDAD | Alta |
| RIESGO | Alta |
| DESCRIPCION | Como administradora debo asignar las fechas que el libro |
| VALIDACION | Quiero poder conocer las fechas tanto de entrega como las de regreso de los libros |
| HISTORIA DE USUARIO # 5 | |
| ID | HU05 |
| NOMBRE | Avisos a tutores de forma fácil y automatizada. (Informe de alumnos deudores). |
| PRIORIDAD | Medio |
| RIESGO | Bajo |
| DESCRIPCION | Como la administradora de la biblioteca debo informar de los deudores |
| VALIDACION | Quiero poder comunicarme inmediatamente con el docente encargado del estudiante para informarle que su estudiante esta de deudor con un libro |

|  |  |
| --- | --- |
| HISTORIA DE USUARIO # 6 | |
| ID | HU06 |
| NOMBRE | Gestión de sanciones para usuarios que no devuelven o pierden artículos |
| PRIORIDAD | Alta |
| RIESGO | Alta |
| DESCRIPCION | Como administradora de la biblioteca debo aplicar una sanción correspondiente a los deudores de los libros |
| VALIDACION | Quiero poder aplicar una sanción a cada estudiante que sea deudor de un libro ya sea económica, horas laborales o veto de la biblioteca |

|  |  |
| --- | --- |
| HISTORIA DE USUARIO #7 | |
| ID | HU07 |
| NOMBRE | Formularios según especificaciones Bibliotecaria |
| PRIORIDAD | Media |
| RIESGO | Bajo |
| DESCRIPCION | Como administradora de la biblioteca debo pedir información al estudiante |
| VALIDACION | Quiero poder pedir el nombre, C.C o T.I, carrera a la que pertenece y se guarde en un lugar que solo pueda ver yo |

DEFINICION DE TAREAS

|  |  |
| --- | --- |
| TAREA | Identificador único para este elemento del proyecto con formato Txx, (siendo xx  numeración progresiva de 01 a 99). |
| HISTORIA DE USUARIO | Historia de usuario a la que pertenece esta tarea. |
| ESTADO | Fase en la que se encuentra el desarrollo de la tarea.  Toma valores: No iniciada, En progreso y Completada. |
| DESCRIPCION | Breve explicación de la finalidad de la tarea. |

TAREAS DEL PROYECTO

|  |  |
| --- | --- |
| TAREA | T01 |
| HISTORIA DE USUARIO | HU07 |
| ESTADO | Realizada |
| DESCRIPCION | Crear formulario de ingreso de información: Se usó HTML y JS para crear la interfaz de usuario. Implementación de frontend. Se Creo una interfaz sencilla y fácil de usar. |

|  |  |
| --- | --- |
| TAREA | T02 |
| HISTORIA DE USUARIO | HU02 |
| ESTADO | Realizada |
| DESCRIPCION | Crea una base de datos con el aplicativo Xampp lenguaje MySQL.  Esta almacenara tanto libros como estudiantes y su información.  Se crea una tabla para almacenar la información del estudiante y una columna donde se agragan los reportes |

|  |  |
| --- | --- |
| TAREA | T03 |
| HISTORIA DE USUARIO | HU06 |
| ESTADO | No será realizada |
| DESCRIPCION |  |
| TAREA | T04 |
| HISTORIA DE USUARIO | HU05 |
| ESTADO | No será realizada |
| DESCRIPCION |  |

|  |  |
| --- | --- |
| TAREA | T05 |
| HISTORIA DE USUARIO | HU03 |
| ESTADO | Realizada |
| DESCRIPCION | Se hace una implementación de js y node y se utilizan como dependencia express para ejecutar el servidor MySQL para integrar la base de datos con Node y Nodemon para ejecutar continuamente el servidor sin reiniciarlo |

|  |  |
| --- | --- |
| TAREA | T03 |
| HISTORIA DE USUARIO | HU07 |
| ESTADO | No será realizada |
| DESCRIPCION |  |

|  |  |
| --- | --- |
| TAREA | T03 |
| HISTORIA DE USUARIO | HU01 |
| ESTADO | No será realizada |
| DESCRIPCION |  |

SPRINT

Para este proyecto emplearemos dos semanas para realizar la entrega de un prototipo funcional del aplicativo para el control de préstamo de libros de la biblioteca FESC.

Se realizará dos sprint con un total de 3 tareas a realizar con su respectiva funcionalidad.

SPRINT1

En el primer SPRINT, se procede a realizar la documentación establecimiento de requisitos y se trabaja en el diseño de la página índex, adicional se establece el formulario y se establecen las tecnologías a integrar. En lapso de la semana se completa la tarea T01 y T02, también se consulta la API que se va a integrar con el Backend.

Daily 1- Daily5

Se logra realizar el diseño del índex en el cual se pudo realizar los avances si ningún contratiempo, se realizan el formulario con los requisitos suministrados por el cliente.

SPRINT 2

En el segundo SPRINT, se procede a diseñar y crear la base de datos localmente, usando XAMPP y MYSQL, se crea la tabla usuarios, se establece la Primary Key, y se estable la columna reportados para establecer la relación de los estudiantes que tengan este estado, adicional se integra el frontend con el Backend, también se valida que se consuma la API seleccionada. Se entrega el producto funcional.

Daily 6-10

En estos Daily´s se procede a tener avances con el Sprint, Se detecta algunas oportunidades de mejora en el Backend, también algunos de los obstáculos que se tuvieron fueron en el consumo de la API que se logra solucionar con el uso de API de google, se detectaron problemas con el Backend en la implementación de rutas.