

INSTITUTO TECNOLÓGICO SUPERIOR DEL OCCIDENTE DEL ESTADO DE HIDALGO

DIVISIÓN DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

Sistema integral web para la gestión, control y seguimiento de residencias profesionales, servicio social y visitas a empresas

Presenta

Carlos Alberto Barrera Lugo

Asesores:

Asesor académico: Sergio Pérez Cruz

TABLA DE CONTENIDO

[I Resumen 3](#_Toc65854748)

[II Abstract 4](#_Toc65854749)

[III Introducción 5](#_Toc65854750)

[IV Planteamiento del problema 7](#_Toc65854751)

[V. Objetivos 8](#_Toc65854752)

[III.1 3.1 General 8](#_Toc65854753)

[III.2 3.2 Específicos 8](#_Toc65854754)

[VI. Justificación 10](#_Toc65854755)

[VII. Delimitación 10](#_Toc65854756)

[VIII. Marco de Referencia 11](#_Toc65854757)

[IX .1 Marco Teórico 12](#_Toc65854758)

[IX.1.1 Teoría general de sistemas (TGS) 12](#_Toc65854759)

[IX.1.2 Ingeniería de sistemas computacionales 13](#_Toc65854760)

[IX.1.3 Ingeniería de software 13](#_Toc65854761)

[IX.1.4 Metodología ágil 13](#_Toc65854762)

[IX.1.5 Sistemas basados en transacciones 14](#_Toc65854763)

[IX.1.6 MVC 14](#_Toc65854764)

[IX.1.7 Clean code 15](#_Toc65854765)

[IX.1.8 Sistema de control de versiones 16](#_Toc65854766)

[IX.1.9 Escalabilidad 16](#_Toc65854767)

[IX.2.0 Refactorización 17](#_Toc65854768)

[IX.2.1 Colección de Datos 17](#_Toc65854769)

[IX.2.2 CRUD 17](#_Toc65854770)

[IX.2.3 Interfaz de usuario 18](#_Toc65854771)

[IX.2.4 Programación Orientada a Objectos 18](#_Toc65854772)

[IX.2.5 PHP 18](#_Toc65854773)

[IX.2.6 MySQL 19](#_Toc65854774)

[IX.2.7 ACL (Access Control List) 19](#_Toc65854775)

[IX.2.8 TCPDF 20](#_Toc65854776)

[IX.2.9 .htaccess 20](#_Toc65854777)

[IX.3.0 Validaciones de datos con expresiones regulares 21](#_Toc65854778)

[IX.3.1 API fetch 22](#_Toc65854779)

[IX.3.2 Conexión PDO 22](#_Toc65854780)

[X. Metodología 23](#_Toc65854781)

[XI Métodos para cada objetivo específico 30](#_Toc65854787)

[XI.1.1 Objetivo 1 30](#_Toc65854788)

[XI.1.2 Objetivo 2 30](#_Toc65854789)

[XI.1.3 Objetivo 3 30](#_Toc65854790)

[XI.1.4 Objetivo 4 30](#_Toc65854791)

[XI.1.5 Objetivo 5 30](#_Toc65854792)

[XII Viabilidad 30](#_Toc65854793)

[XII.1.1 Técnica 31](#_Toc65854794)

[XII.1.2 Económica 31](#_Toc65854795)

[XII.1.3 Operativa 32](#_Toc65854796)

[XIII. Resultados 33](#_Toc65854799)

[XIV Conclusiones 41](#_Toc65854801)

[XV Recomendaciones 41](#_Toc65854802)

[XVI Bibliografía 42](#_Toc65854803)

[XVII Anexos 45](#_Toc65854804)

[XVII.1 Manual de usuario 45](#_Toc65854805)

# Resumen

Las Tecnologías de la Información y Comunicación (TIC) y los sistemas de Información Basados en Web han cambiado la forma en que operan las instituciones; el crecimiento tecnológico ha permitido que las instituciones educativas cuenten con sistemas informáticos de forma oportuna; cuya incorporación logran mejoras significativas; El Instituto Superior del Occidente del Estado de Hidalgo (ITSOEH) es una Institución de Educación Superior en el estado que imparte 8 carreras a nivel licenciatura y mediante el departamento de Control Escolar y Vinculación realiza la gestión y vinculación con los sectores productivos, permitiendo la integración del estudiante a la actividad laboral capases de responder a las necesidades de la sociedad; por lo anterior el proyecto Sistema integral web para la gestión, control y seguimiento de residencias profesionales, servicio social y visitas a empresas, tiene como objetivo eficientar las actividades administrativas del departamento, el control y seguimiento de expedientes y trámites de los estudiantes; contiene los módulos de residente, servicio social, catálogo de empresas, asesor de residencia, docentes, reportes, estadísticas y notificaciones. El sistema web se basó en el modelo de desarrollo Ágil sin la utilización de algún frameworks, empleando PHP como lenguaje.

Todos los datos y formatos obtenidos y generados de los procesos anteriormente mencionados se almacenarán en una base de datos en MySQL, con ello se plantea dar un cambio a la arquitectura en la que la institución lleva su control de los alumnos, dado a que solo utilizan tablas en Excel para los registros.

Las actividades realizadas para el desarrollo del proyecto se llevaron a cabo acorde a un plan de trabajo establecido en base a la metodología mencionada anteriormente.

**Sistemas web, servicio social, residencia profesional, estudiante, asesor, sistemas de gestión educativa.**

# Abstract

Information and Communication Technologies (ICT) and Web-Based Information systems have changed the way institutions operate; technological growth has allowed educational institutions to have computer systems in a timely manner; whose incorporation achieve significant improvements; The Higher Institute of the West of the State of Hidalgo (ITSOEH) is a Higher Education Institution in the state that teaches 8 undergraduate degree courses and through the Department of School Control and Linkage, it manages and links with the productive sectors, allowing integration from the student to the work activity capable of responding to the needs of society; Therefore, the project Comprehensive web system for the management, control and monitoring of professional residencies, social service and company visits, aims to streamline the administrative activities of the department, control and monitoring of student files and procedures; It contains the modules of resident, social service, company catalog, residence advisor, teachers, reports, statistics and notifications. The web system was based on the Agile development model without the use of any frameworks, using PHP as the language.

All the data and formats obtained and generated from the aforementioned processes will be stored in a database in MySQL, with this it is proposed to give a change to the architecture in which the institution takes control of the students, since they only use tables in Excel for the records.

The activities carried out for the development of the project were carried out according to a work plan established based on the methodology mentioned above.

**Web systems, social service, professional residence, student, advisor, educational management systems.**

# Introducción

El proyecto desarrollado surge de la necesidad del Departamento de Control Escolar y Vinculación, que tiene como función el de planear, coordinar, controlar y evaluar las actividades relacionadas con las prácticas y promoción profesional, servicio social y desarrollo comunitario, así como la vinculación con los sectores productivos, permitiendo la integración del estudiante a la actividad laboral para la gestión de expedientes y trámites de los estudiantes de todas las carreras que realizan su residencia profesional, servicio social y visitas a empresas por semestre.

De acuerdo al Lineamiento para la Operación y Acreditación de la Residencia Profesional versión 1.0 Planes de Estudio 2009- 2010, Fecha de creación 27 de octubre de 2011; Una residencia profesional es una estrategia educativa con carácter curricular que permite al estudiante aún en proceso de formación, incorporarse profesionalmente al sector productivo de bienes y servicios a través del desarrollo de un proyecto asesorado por instancias académicas y externas; y al Lineamiento para la Operación y Acreditación del Servicio Social versión 1.0 Planes de Estudio 2009-2010, Fecha de creación 2 de septiembre de 2011 el servicio social tiene la finalidad de fortalecer en el estudiante la formación integral. Desarrollando una conciencia de solidaridad, compromiso y respeto hacia la sociedad en la que se desenvuelve; por lo que el desarrollo del Sistema integral web para la gestión, control y seguimiento de residencias profesionales, servicio social y visitas a empresas, cumplirá con los periodos y formatos establecidos en bajo los lineamientos del Tecnológico Nacional de México y tiene como objetivo automatizar las actividades administrativas.

El control y seguimiento de expedientes de cada estudiante que se encuentre realizando la residencia profesional dentro y fuera del estado o el servicio social; cada residente o prestador de servicio conoce de manera oportuna en qué parte del trámite se encuentra, ya que el sistema lleva un seguimiento de cada proceso, y es notificado a través del correo electrónico sobre tiempos de entrega de formatos, documentos y reportes; una vez concluido el proceso, el sistema genera las constancias de liberación en formato pdf; en relación a la visita de empresas cuenta con un catálogo de todas las empresas disponibles para recibir visitas, y una vez aprobada la solicitud por parte del docente y del asesor industrial, puede visualizar el estado del trámite de la visita. .

Dentro del mismo documento se encuentran explicados los pasos de la metodología utilizada, pasos que permitieron que el sistema se realizara de una forma más ordenada, con lo establecido en la metodología se pudo completar los objetivos establecidos, permitiendo que el sistema sea adaptable a futuras mejoras si la institución lo requiere, también al haberse realizado siguiendo esos puntos establecidos se pueden identificar las partes que componen el sistema, lo que facilita el trabajo de mitigar fallas en caso de que se produzcan.

También se analizará la factibilidad del sistema propuesto en este documento, tomando en cuenta los factores y las condiciones en las que se desarrollan las actividades para el proceso del servicio social y residencia profesional, buscando una forma de que el sistema pueda satisfacer las necesidades de los usuarios y con una posibilidad de ampliarse no solo a el proceso del servicio social, permitiéndole en un futuro agilizar el proceso de la residencia profesional.

# IV Planteamiento del problema

El Departamento de Control Escolar y Vinculación cuenta con tres áreas: Residencias profesionales, servicio social y visitas a empresas. Cada una de las áreas lleva el control de los expedientes y trámites de los estudiantes que realizan la residencia profesional, servicio social y visitas a empresas respectivamente; sin embargo se dificulta el manejo de expedientes de cada estudiante debido a que se realiza de forma manual, no se tiene un control para la entrega de documentación de los estudiantes, tampoco para emitir e informar al estudiante que puede pasar a recoger o entregar algún documento que tenga pendiente; los documentos son requisitados de forma manual, esto provoca pérdida de tiempo en caso de errores, no se cuenta con el seguimiento de reportes parciales y finales por lo que no se cumplen con las fechas de entrega estipuladas.

Finalmente en el área de visitas a empresas, no se tiene un control de las visitas que realizan los docentes con sus estudiantes a las diferentes empresas; no se notifican a los docentes de las fechas en que se permiten realizar las solicitudes para los viajes, y en ocasiones lo hacen fuera de tiempo lo que provoca que no puedan llevarse a cabo, en todos los procesos de búsqueda de expedientes se produce una pérdida de tiempo considerable al realizarse de forma manual, dado a que llevan a cabo sus formularios y controles utilizando plataformas externas al instituto, así mismo no cuentan con un gestor de datos para almacenar la información de manera digital, haciendo que se tengan que imprimir los formatos.

Se optó por utilizar un sistema web dado a que el avance de las telecomunicaciones ha permitido ampliar el alcance de este tipo de sitios, con ello los usuarios podrían acceder a al sistema desde cualquier equipo con conexión a internet, el sitio también permite a los alumnos mantenerse al tanto de las fechas para realizar sus registros, con el fin de evitar posibles confusiones y conflictos a la hora de hacer sus registros y actividades.

# Objetivos

## 3.1 General

Desarrollar un sistema web para el Tecnológico ITSOEH que permita tener el control de los registros y reportes de actividades de los alumnos que realizan su servicio social y residencia profesional, limitándose únicamente a la comunidad perteneciente al instituto y que cumpla las condiciones para iniciar su servicio social.

## 3.2 Específicos

* Identificar los procedimientos que deben realizar los alumnos para realizar su registro.
* Contemplar los distintos formatos que deben llenar.
* Conocer los periodos para realizar las convocatorias de inscripción, fechas de evaluaciones y fechas de visita a empresas.
* Delimitar las condiciones para que se pueda realizar el registro al sistema.
* Asignar roles de usuario dentro del sistema, para limitar la navegación dentro del sistema.
* Permitir la generación y llenado de formatos dentro del sistema.
* Implementar el uso de una base de datos para almacenar la información de los alumnos y sus documentos.
* Crear una página de inicio de sesión.
* Implementar sistemas de CRUD dentro del sistema, lo que permitirá que los usuarios puedan manipular la base de datos.
* Limitar las acciones de los usuarios en base al rol asignado
* Establecer fechas dentro del sistema que habiliten los registros y las evaluaciones en base a la convocatoria de inscripción
* Almacenar los formatos dentro del sistema para que los alumnos puedan acceder a ellos.
* Validar las inscripciones al servicio social y residencia profecional en base a las condiciones que se establezcan en la convocatoria.
* Establecer una secuencia a los procesos para evitar errores y confusiones al momento de realizar las inscripciones y subir evaluaciones.
* Crear un módulo donde el alumno pueda subir sus evaluaciones y formatos.
* Dar a conocer el estatus del proceso en el que se encuentra el alumno.
* Gestionar el estado de los alumnos, para llevar un control correspondiente a sus carreras y a los sitios donde realicen su servicio social
* Generar las constancias de liberación, una vez que el alumno haya concluido con el servicio social.

# Justificación

Dado a que el personal a cargo de gestionar el proceso de servicio social es muy pequeño, se han visto con una carga de trabajo excesiva en muchas ocasiones ya que al no contar con un sistema definido a la hora de realizar los registros y gestionar los datos de los alumnos, se veían en un conflicto ya que los alumnos en muchas ocasiones no respetaban los pasos para realizar sus procesos o desconocían las fechas en las que debían entregar sus actividades, como se menciona anteriormente, el personal a cargo es muy pequeño mientras que los alumnos que se registran son aproximadamente 300, dado a que los periodos de evaluación consta de menos de una semana, el personal debe revisar todos los formatos siguiendo un orden, sin embargo en el mayor de casos los alumnos no saben cómo es el orden los formatos y sus lineamientos, lo cual provoca que las revisiones que se llevan a cabo sean redundantes y por ende se obtiene un derroche de tiempo en revisar constantemente a los alumnos que aún no cumplen con todos los lineamientos para sus registros y evaluaciones.

# Delimitación

El proyecto consta de un periodo de 6 meses para concluirse, dado a que solo se plantea dirigirlo a la comunidad interna del instituto no supone manejar más de 500 usuarios a la vez, por lo que se estima que será utilizado por un rango determinado de usuarios en ciertos periodos de tiempo, así mismo se contempla un almacenamiento temporal de los datos que se almacenan en el servidor del sistema.

Se identificaron diferentes tipos de usuarios con sus respectivos roles y privilegios para el acceso al sistema. El usuario Administrador, tiene el acceso total al sistema web y a los perfiles de cada usuario, actualización, respaldo de la base de datos y mantenimiento a los módulos; los usuarios: Jefe del Departamento de Control Escolar y Vinculación, Jefe de Carrera, Asesor Académico, pueden crear, editar, registrar, guardar, consultar, filtrar, buscar, dar de baja, eliminar, residentes, empresas, convenios, visualizar el avance de las actividades.

Es decir, en qué parte del proceso se encuentra el residente o prestador de servicio o en su defecto si tiene algún documento pendiente por entregar, e imprimir las constancias. El Usuario Jefe de Control Escolar puede consultar, visualizar, y buscar en el catálogo de proyectos; El usuario Asesor Académico y estudiante pueden consultar, imprimir, recibir notificaciones, descargar los documentos y constancias en formato PDF.

Para llevar a cabo la implementación del software será necesario realizar una serie de pruebas funcionales y considerar la retroalimentación de los usuarios para adaptar el programa a sus necesidades, este proyecto solamente será utilizado por la línea de producción y su funcionamiento está limitado a la visualización de información.

En cuanto a aspectos socio-administrativos se sabe que la gerencia no necesitará incurrir en gastos extras para la implementación, pues el esfuerzo más grande recae en cómo se coordinarán las pruebas con el personal.

Los proyectos que se tienen planeados implementar están delimitados por el tiempo en que se cuente con personal de residencia profesional para elaborar el software, en la (ilustración 1) se muestra detalle cómo se tiene organizado el tiempo para desarrollar los productos.

*(ilustración 1 Cronograma pendiente)*

# Marco de Referencia

En este apartado se encuentra la base de información que sustenta el desarrollo tecnológico de este proyecto, pasando por bases teóricas relacionadas al desarrollo del proyecto, posteriormente se enlistarán algunas referencias a proyectos que antecedan al presente sirviendo de base para la selección de tecnologías y prácticas tanto en el desarrollo como la gestión. Finalmente se hará la respectiva mención de cada concepto para ayudar a mejorar la compresión al lector.

# .1 Marco Teórico

En este apartado se encontrarán los detalles acerca de las teorías en donde se fundamente la investigación para la realización del presente proyecto, así mismo también se mencionarán los conceptos puntuales que dan origen al desarrollo del sistema y que son realmente esenciales para comprender el objetivo de llevar a cabo la implementación del software se comenzará en detallar aspectos generales hasta ir a conceptos puntuales.

### IX.1.1 Teoría general de sistemas (TGS)

La observación de un concepto fundamental en un área determinada varía de acuerdo a los objetivos de la misma, es decir que cada objeto de estudio derivará de un elemento en particular, por ejemplo. Una célula es la base para la biología, un dato para la informática e incluso el pensamiento es la base para estudiar la lógica y el razonamiento. Para representar algunos tópicos complejos de áreas especializadas de la ciencia se volvió indispensable el uso de un modelo de construcción teórica que pueda establecer un nivel intermedio entre un contenido específico (que aporte poco o nulo significado general) y la parte más general (que pueda carecer de contenido) (Johansen Bertoglio, 1993). La presente investigación se sustenta en la TGS debido a que el aplicativo construido formará parte de un sistema de información, formalmente un TGS pretende ser una forma sistemática y científica de aproximación y representación de la realidad, al mismo tiempo, como una orientación hacia una práctica estimulante para formas de trabajo transdisciplinarias (Arnold Cathalifaud & Osorio, 1998).

### IX.1.2 Ingeniería de sistemas computacionales

La TGS funge como un esqueleto para el estudio de las ramas de la ciencia y dentro de estas se encuentran diversas tendencias que apuntan a un contexto aplicativo combinando el pensamiento ingenieril se dio origen a dicha rama interdisciplinar, hace referencia a la planeación, diseño, evaluación y construcción científica de sistemas hombre – máquina (Johansen Bertoglio, 1993). Debido al cambio generacional y a la aparición de los primeros ordenadores surgió una vertiente en la rama de la ingeniería de sistemas que involucraba precisamente las ciencias de la computación adquiriendo los principios de creación hacia elementos no tangibles pero claramente visibles, este apartado teórico es la base fundamental para la creación ya sea de un sistema o parte del mismo basado en los elementos de las ciencias de la computación como lo son la programación, paradigmas, estructuras de datos, arquitectura, entre otros elementos detallados más adelante.

### IX.1.3 Ingeniería de software

Esta rama de las ciencias de la computación aporta un enfoque orientado a la construcción de aplicaciones, es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales (Pressman, 2010). Otra definición que abarca más aspectos es aquella desarrollada por la IEEE en la que se dice la ingeniería de software es la aplicación de un enfoque sistémico, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir la aplicación de la ingeniería al software (The Institute of Electrical and Electronics Engineers, 1990), se puede deducir que la repercusión de este campo de la ingeniería impacta en la forma de gestionar los recursos para llevar a cabo la concepción del software.

### IX.1.4 Metodología ágil

La metodología de trabaja que se emplea en el desarrollo de software robusto está orientado a modelos estructurados y bien documentados, que en un principio funcionan para equipos expertos y muy bien coordinados, pero cuando se trata de ciclos de desarrollo cortos estas estructuras clásicas de desarrollo presentan ciertos problemas, las metodologías ágiles provienen del manifiesto ágil y pretende optimizar el proceso de creación de software, se valora entonces:

* Al individuo y sus interacciones del equipo de desarrollo sobre el proceso y las herramientas.
* Desarrollar software que funcione más que la documentación del mismo.
* La colaboración con el cliente más que la negociación de su contrato.
* Responde a los cambios más que seguir con el plan establecido.

Además, se siguen doce principios para seguir realmente una metodología ágil (López Menéndez de Jiménez, 2015). Dentro de las metodologías ágiles se encuentran Kanban y LEAN como metodologías menos específicas y por tanto más fáciles de adaptar a un equipo de trabajo con relativa facilidad, dentro de este proyecto se pretende adoptar principios de ambas metodologías, especialmente de Kanban, pero sin seguir pautas rígidas (Project Management Institute, 2017).

### IX.1.5 Sistemas basados en transacciones

Son sistemas generalmente localizados en un ordenador específico el cual permite llevar registrar y llevar a cabo transacciones (consultas, altas y bajas) que son necesarias para el desarrollo de una empresa, el objetivo principal de este tipo de sistemas es automatizar el procesamiento de los volúmenes de información mismos que se almacenan en una base de datos o archivos específicos de colecciones (Pítsica Marques, 2001). El proyecto que se pretende llevar a cabo está basado en este tipo de sistema de información debido a que ayuda a automatizar un proceso de visualización de datos con ayuda de un servidor para realizar operaciones de tipo CRUD en un archivo específico evitando la redundancia del guardado de información.

### IX.1.6 MVC

El desarrollo de un producto de software tiene una implicación muy importante en la estructura o patrón de diseño para implementar una solución, según la Universitat d’Alacant (Universidad de Alicante), el Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

### IX.1.7 Clean code

Es una serie de pensamientos que tratan de dar un estándar adecuado para generar una alta mantenibilidad del código que se desarrolle, esta manera de desarrollar código permite obtener beneficios como la facilidad de desarrollar pruebas sean del tipo que sean, si se adoptan este tipo de medidas además será mucho más sencillo que cualquier otra persona pueda comprender que está pasando en el software (Suryanarayana, Samarthyam, & Sharma, 2015). Las prácticas de clean code no solo abordan como evitar malas prácticas si no que a su vez se abordan temas de deuda técnica o código hediondo que pudieran comprometer el desarrollo futuro del código, dentro de la aplicación que se trabajó se detectó una deuda técnica orientada a: el diseño, pues carecía de un estándar y todo parecía ejecutarse en una misma clase que combinaba la parte de la interfaz de usuario, el código, debido a que se encontraban múltiples carencias sobre el conocimiento del lenguaje, estándares para nombrar elementos del código y carencia de estructuras de datos para manipular la información, en cuanto a la documentación se encontró un gran déficit, pues no existía una manera de conocer cómo funcionaba cada parte importante del código ya que se incluían únicamente comentarios relacionados a partes fáciles de comprender e incluso un tanto obvias.

### IX.1.8 Sistema de control de versiones

Una parte crucial para llevar un mejor control sobre el desarrollo del proyecto es cuando se implementa un sistema de control de versiones, pues este tipo de sistema puede registrar los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo (Git, 2014) si se realiza un análisis más profundo sobre cómo funciona el control de versiones, se puede notar que existen diversos tipos de sistemas de entre los cuales destaca uno principalmente por su alta disponibilidad y capacidad de colaboración este es denominado como *sistema de control de versiones distribuido* en el cual se basa Git el cuál destaca debido a que cuenta con una alta demanda por la comunidad de desarrolladores alrededor del mundo debido a que no controla cada cambio específico sino que más bien hace una especie de captura de todos los archivos cada vez que se guardan se optó por utilizar esta tecnología para llevar el control del proyecto debido a que la colaboración aunada con GitHub puede ser altamente eficiente para trabajar en equipo.

### IX.1.9 Escalabilidad

Es importante recalcar que una aplicación sea pequeña o grande en el momento que se despliega pasados algunos años pueden surgir más necesidades para resolver y es el deber del equipo de desarrollo dejar un sistema escalable. La escalabilidad es la habilidad que tiene un sistema para para adaptarse al crecimiento sin perder calidad en los servicios que se ofrecen, es de suma importancia mencionar que la capacidad de crecimiento puede darse de dos maneras: scale in (escalado vertical) que hace referencia al crecimiento en un solo nodo, por otro lado, el scale out (escalado horizontal) que se enfoca en el aumento de nodos para liberar la carga del sistema (Gonzalez Godino, 2013)

### IX.2.0 Refactorización

Cuando se diseña el software normalmente se debería pensar en crearlo orientado hacia la mantenibilidad del código fuente, una de las principales reglas para llevar a cabo este cometido se trata de escribir partes simples para lograr un cambio en la estructura interna para lograr que este sea más sencillo de comprender y sobre todo fácil de modificar sin alterar el comportamiento normal. Generalmente las técnicas de refactorización de código se aplican después de que el funcionamiento de un módulo ha sido desarrollado, pasa el tiempo y al revisar el código se encuentra una difícil comprensión del funcionamiento o simplemente el producto ha crecido y necesita una mejor forma de escalar (Dooley, 2017).

### IX.2.1 Colección de Datos

Para que un programa cuente con una alta eficiencia debe incluirse un buen manejo de la memoria aplicando estructuras de datos o colecciones de datos que no son más que un conjunto de técnicas que aumentan considerablemente la productividad del programa, reduciendo en elevado grado, el tiempo requerido para escribir, verificar, depurar y mantener los programas. El término estructura de datos hace referencia a un conjunto de datos que, por medio de un nombre, identifican un espacio en memoria, teniendo ciertas características como la organización y estructuración, permitiendo realizar operaciones definidas en ellas. Las estructuras de datos pueden ser de dos tipos: estructuras de datos estáticas (aquellas que cuentan con un tamaño definido), estructuras de datos dinámicas (aquellas en las que el tamaño puede variar en tiempo de ejecución) (Luna López, 2012).

### IX.2.2 CRUD

Es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos o un almacén de datos este acrónimo deriva de las palabras: Create (Crear), Read (Leer), Update (Actualizar) y Delete (Eliminar) (Mozilla and contribuitors, 2019).

### IX.2.3 Interfaz de usuario

Para que un usuario ordinario se pueda comunicar con un ordenador o aplicación del ordenador hace falta contar con un conjunto de elementos de hardware y software que presenten la información de una forma entendible apoyándose de gráficos, colores, tipografía presentados en forma de ayuda, manuales, tutoriales, etc. Si la interfaz de usuario está perfectamente diseñada el usuario podrá navegar por la aplicación u ordenador sin problema alguno encontrando las respuestas a cada acción que realice (Sebastián Gómez, 2000).

### IX.2.4 Programación Orientada a Objectos

Conocido como un paradigma de programación, son un conjunto de técnicas que nos permiten incrementar enormemente nuestro proceso de producción de software; aumentando drásticamente nuestra productividad por un lado y permitiéndonos abordar proyectos de mucha mayor envergadura (Morero, 2000), la POO se encarga de definir un objeto como una serie de métodos y atributos que se encontrarán definidos en una clase (abstracción de la realidad, sirve como una especie de plano) al utilizar un paradigma de programación orientado a objetos es posible observar un grado de reusabilidad en el código que van de la mano con otra serie de conceptos que son:

1. **Herencia:** Destaca la cualidad de contar con clases padre e hijo para reutilizar elementos de las clases derivadas.
2. **Encapsulación:** Cuando se reúnen todos los elementos pertenecientes a una misma entidad.
3. **Polimorfismo:** Comportamientos similares asociados a objetos distintos que pueden compartir este funcionamiento con elementos distintivos de cada uno.

### IX.2.5 PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

### IX.2.6 MySQL

A medida que las aplicaciones informáticas comenzaron a popularizarse, se observó la vertiente de contar con datos redundantes para un conjunto de aplicaciones específico, por ejemplo un nombre de producto podría estar en el inventario y al mismo tiempo en el detalle de venta desde ese momento fue necesario comenzar a utilizar conjuntos de ficheros interrelacionados que implementaban estructuras complejas que a su vez, podían ser compartidos por varios procesos de manera simultánea a este descubrimiento se le conoce como bases de datos, de una manera más formal se puede decir que una base de datos es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del sistema de información y de sus interrelaciones.

### IX.2.7 ACL (Access Control List)

Manejar permisos y roles para nuestros usuarios dentro de una aplicación nos da una gran flexibilidad a la hora de tomar decisiones sobre a qué grupo de usuarios debemos mostrar algún tipo de contenido y a quienes debemos ocultárselos. Si tienes algo de experiencia o tiempo en este mundo del desarrollo quizás ya sepas algo sobre ALC (Fernandes, s.f.)

Access Control List no es más que un concepto de seguridad informática el cual es usado para separar los privilegios o permisos de nuestra aplicación. Podemos trabajar con un sistema de roles o simplemente asignando a los usuarios cuáles van a ser sus permisos dentro de nuestra aplicación.

### IX.2.8 TCPDF

TCPDF es una Open Source Clase/Biblioteca para el Popular Lenguaje de Programación Web PHP v4 y v5, la cual permite crear ficheros PDF al vuelo, es decir dinámicamente. Dos de las cualidades más apreciadas de esta clase, es su simplicidad a la hora de crear archivos PDF y la capacidad de interpretar código XHTML.

### IX.2.9 .htaccess

Un fichero .htaccess (hypertext access), también conocido como archivo de configuración distribuida, es un fichero especial, popularizado por el Servidor HTTP Apache que permite definir diferentes directivas de configuración para cada directorio (con sus respectivos subdirectorios) sin necesidad de editar el archivo de configuración principal de Apache

El fichero .htaccess ofrece un universo de posibilidades, detallamos a continuación los usos más frecuentes.

* **Autorización y autenticación**

Los ficheros .htaccess son usados frecuentemente para especificar restricciones de seguridad para un directorio en particular, de aquí el sufijo "access". El fichero .htaccess se acompaña frecuentemente de otro fichero .htpasswd que guarda usuarios válidos y sus contraseñas.

* **Creación de URLs Amigables (semánticas)**

Los servidores suelen usar el .htaccess para reescribir URLs largas y complejas, en otras más simples y fácilmente recordables.

* **Restringir el acceso**

Permite bloquear (usando allow/deny) usuarios por su dirección IP y/o dominio e ISPs. También permite bloquear bots y arañas web.

* **Server Side Includes**

Permite el conjunto de directivas Server Side Includes (SSI).

* **Listado de directorio**

Controlar el comportamiento por defecto del servidor cuando no se especifica ninguna página web.

* **Crear redirecciones estáticas**
  + **Página de error personalizada**

Cambiar la página que se muestra cuando un error de la parte del servidor ocurre, por ejemplo: HTTP 404 Not Found.

* **Multipurpose Internet Mail Extensions (MIME) types**

Controlar cómo el servidor maneja diferentes tipos de archivos.

* **Control de caché**

Los ficheros .htaccess permiten a un servidor controlar la caché web por medio de los navegadores web y proxis para reducir el uso del ancho de banda, la carga del servidor, y el lag percibido.

* **Evitar hot-linking.**
* **Forzar dominio sin WWW**
* **Evitar envíos de paquetes PING**

### IX.3.0 Validaciones de datos con expresiones regulares

Una expresión regular o regex es un patrón que se compara con una cadena objetivo de izquierda a derecha, carácter a carácter. La biblioteca PCRE (Perl Compatible Regular Expressions) es una extensión incorporada en PHP que permite utilizar expresiones regulares en funciones para buscar, comparar y sustituir strings en PHP.

### IX.3.1 API fetch

La API Fetch proporciona una interfaz JavaScript para acceder y manipular partes del canal HTTP, tales como peticiones y respuestas. También provee un método global fetch() que proporciona una forma fácil y lógica de obtener recursos de forma asíncrona por la red.

Este tipo de funcionalidad se conseguía previamente haciendo uso de XMLHttpRequest. Fetch proporciona una alternativa mejor que puede ser empleada fácilmente por otras tecnologías como Service Workers. Fetch también aporta un único lugar lógico en el que definir otros conceptos relacionados con HTTP como CORS y extensiones para HTTP.

La especificación fetch difiere de JQuery.ajax() en dos formas principales:

* El objeto Promise devuelto desde fetch() no será rechazado con un estado de error HTTP incluso si la respuesta es un error HTTP 404 o 500. En cambio, este se resolverá normalmente (con un estado ok configurado a false), y este solo será rechazado ante un fallo de red o si algo impidió completar la solicitud.
* Por defecto, fetch no enviará ni recibirá cookies del servidor, resultando en peticiones no autenticadas si el sitio permite mantener una sesión de usuario (para mandar cookies, credentials de la opción init deberán ser configuradas). Desde el 25 de agosto de 2017. La especificación cambió la política por defecto de las credenciales a same-origin. Firefox cambió desde la versión 61.0b13..

### IX.3.2 Conexión PDO

PDO (PHP Data Object) es una interfaz ligera y consistente para acceder a bases de datos en PHP de forma estándar, sin utilizar los métodos propios de cada gestor de bases de datos. De esta forma se consigue que podamos usar el mismo programa PHP para conectar con cualquier base de datos.

**Parámetros de la conexión**

Primero debemos declarar una instancia de un objeto de la clase PDO para manejar la conexión. Al objeto debemos pasarle los parámetros de la conexión:

1. DSN, DB\_USER, DB\_PASS y DB\_OPTIONS.
2. DSN Orígen de los datos. Indica el controlador de la base de datos y los parámetros de la conexión: Ip del servidor y nombre de la BD.
3. DB\_USER Nombre de usuario de la base de datos.
4. DB\_PASS Contraseña de la base de datos para ese usuario.
5. DB\_OPTIONS Opciones de conexión (opcional).

# Metodología

En este aparatado se explica a detalle la organización de recursos, requerimientos y procedimientos que permiten realizar el presente proyecto de desarrollo del sistema web, en un principio se mencionará sobre el tipo de metodología ágil, que beneficios trae al desarrollo que hacen optimo su uso en contra de una metodología tradicional, posteriormente se enlistarán algunas metodologías de las cuales se han obtenido elementos específicos y porque no se ha optado por seguir una metodología al pie de la letra, en el apartado final se hará mención del proceso que se ha seguido con ilustraciones que ayuden a corroborar este proceso y además se incluye un apartado sobre la viabilidad técnica, económica y operativa.

Cuando se piensa en comenzar el desarrollo de un proyecto, existen diversas maneras de atacar el problema, generalmente un programador optaría por desarrollar una solución lo antes posible para realizar cambios y ajustarse a las necesidades del cliente esto quiere decir que se enfoca a los requerimientos únicamente e ignora por completo aspectos como el tiempo, costo y procedimientos externos involucrados. En caso de que una persona que gestiona proyectos que no son precisamente de software pretenda llevarlo a cabo se notará una mejora en el control de recursos y sabrá diferenciar que procesos externos intervienen, sin embargo, la implementación de cambios en los requerimientos le supondrá un reto enorme pues no está acostumbrado a los cambios repentinos y su gestión del proyecto podría tomar el doble de tiempo si comenzara a realizar esos cambios desde un inicio.

De acuerdo con Pressman (2010), la ingeniería de software proporciona los elementos necesarios para llevar un enfoque no solo eficaz, sino que también eficiente pues se centra en aportar principios fundamentales divididos en principios que guían el proceso y principios que guían la práctica (ver tabla 1).

Tabla 1. Principios fundamentales de la ingeniería de software (elaboración propia).

|  |  |  |
| --- | --- | --- |
|  | Principios que guían el proceso | Principios que guían la práctica |
| Significado | Establece un fundamento filosófico para guiar actividades estructurales | Definen un conjunto de valores y reglas que sirven para analizar un problema, diseñar una solución, implementar, probar y entregar. |
| Principios | **Ser ágil.** Mantener un enfoque técnico sencillo tanto como sea posible, generar productos concisos, tomar decisiones locales en la medida de lo posible | **Divide y vencerás.** Enfatizar la separación de entidades, un problema grande se vuelve más sencillo si se divide en pequeñas entidades. |
| **Centrarse en la calidad.** Aplica para toda actividad, acción y/o tarea del proceso | **Entender el uso de la abstracción.** Conocer primero los detalles antes de utilizar la abstracción en un nivel alto o bajo. |
| **Estar listo para adaptar.** Es normal cambiar el enfoque a las restricciones impuestas, no hay una fórmula para hacer las cosas | **Buscar coherencia.** Aplicar un contexto familiar hace que el software sea más fácil de usar. |
| **Formar un equipo eficaz.** Un equipo de confianza, organizado y comunicado. | **Centrarse en la transferencia de información.** Poner atención especial al flujo de la información dentro del análisis, diseño, construcción y prueba. |
| **Establecer mecanismos para la comunicación y coordinación.** Es de vital importancia para que el proyecto no falle | **Construir software con modularidad eficaz.** Centrar cada módulo en un aspecto delimitad del sistema y su conexión debe ser sencilla. |
| **Administrar el cambio.** Deben establecerse mecanismos para administrar la forma en que los cambios se solicitan, evalúan, aprueban e implementan. | **Buscar patrones.** Ayudan a formar un cúmulo de información para futuras referencias. |
| **Evaluar el riesgo.** Es necesario establecer planes de contingencia. | **Cuando sea posible representar el problema y su solución desde varias perspectivas.** Es más probable que se tenga mayor visión, lo que ayuda a detectar errores y omisiones. |
| **Crear productos con valor para otras actividades.** El producto de trabajo imparte la información necesaria sin omisiones ni ambigüedades. | **Tener en mente que alguien dará mantenimiento al software.** Es importante aplicar técnicas de ingeniería de software para asegurar que el software tenga mantenimiento y no una refactorización completa. |

Independientemente si se utiliza una metodología ágil o tradicional, estos principios deben estar parcialmente cubiertos especialmente cuando se combinan aspectos de distintas metodologías.

La metodología de la que parte el proyecto es una serie de elementos de kanban, una metodología ágil caracterizada por contar con una rápida adopción en proyectos en donde el equipo de desarrollo no supera los 10 integrantes (incluso puede ser usada para una sola persona), kanban generalmente está orientado a mejorar el control del flujo de trabajo, según Anderson & Carmichel (2016) Kanban sigue el modeo de pensamiento de “*Comienza por donde estés*” siendo así una metodología centrada en el cambio rápido y focalizado.

Para hacer uso de esta metodología se utiliza un sistema kanban el cual permite controlar el flujo de entrega limitando la cantidad de trabajo en progreso con ayuda de una serie de tarjetas denominadas kanbans distribuidas en un tablero kanban (ver ilustración 2) que no es más que una pizarra digital o física con una serie de divisiones que no están fuertemente definidas siempre y cuando existan los elementos esenciales de Kanban.

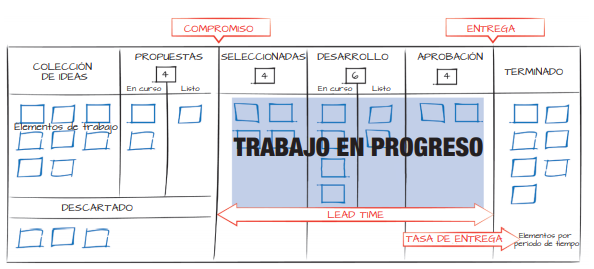


Ilustración 2. Ejemplo de tablero kanban (extraído de Anderson & Carmichel, 2016).

* **Reconocer el trabajo a futuro:** Que requerimientos son los solicitados, como serán implementados y si existen casos en los que no se atenderán esos requerimientos por diversos motivos.
* **Identificar el trabajo en progreso:** Una serie de columnas en donde se expresa que trabajo está realizando cada miembro del equipo.
* **Establecer políticas:** En este apartado se incluye la delimitación del WiP (Work in Progress), nivelación de carga, definición de columnas y políticas de retroalimentación.

Se adoptaron estos principios para la gestión del proyecto debido al enfoque iterativo, incremental y orientado a la mentalidad del manifesto ágil se optó por tomar elementos importantes de Kanban sin seguir una pauta estricta del mismo, a continuación se puede observar la implementación de un tablero kanban tradicional con ayuda de la herramienta Trello (ver ilustración 3).

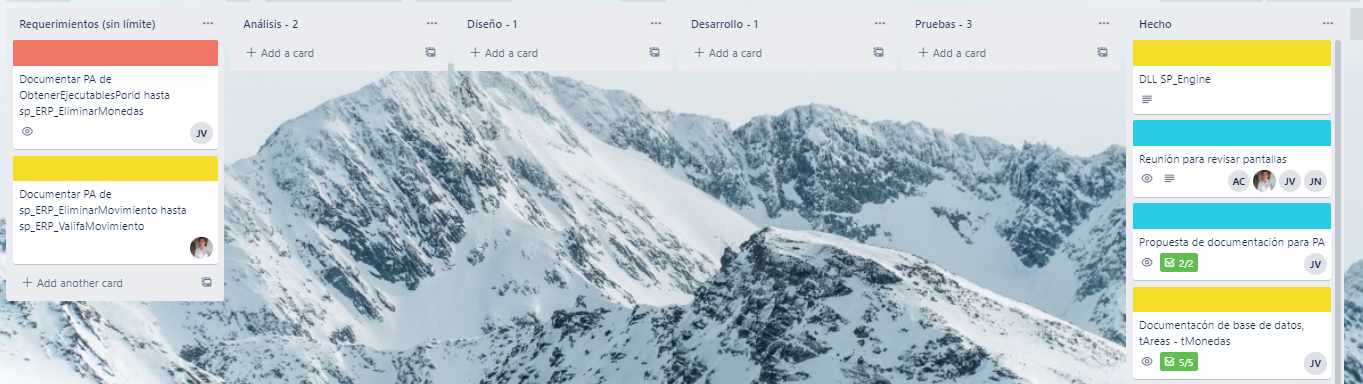


Ilustración 3. Tablero kanban para el proyecto (ilustración propia).

Los elementos de las columnas están organizados de la siguiente manera:

* **Requerimientos (sin límite):** En este apartado se colocan los nuevos requerimientos que vayan surgiendo y se analiza que tan viable es implementarlos o posponerlos.
* **Análisis (2):** Para esta columna solo se pueden colocar dos elementos en progreso, debido a que como el equipo está conformado por dos desarrolladores supervisados por un CTO el análisis puede estar dirigido únicamente por alguno de los dos desarrolladores apoyado en el CTO.
* **Diseño (1):** Uno de los desarrolladores está enfocado mayormente en el tema de frontend (aplicaciones del lado del servidor) y cuenta con las habilidades necesarias sobre desarrollo de interfaces de usuario.
* **Desarrollo (1):** Debido a que el miembro del equipo restante basa su experiencia en aplicaciones para el backend solo puede existir un trabajo para la columna del desarrollo en cada momento de tiempo.
* **Pruebas (3):** En esta sección se coloca un mayor WiP debido a que además de los desarrolladores el CTO puede apoyar en la tarea de pruebas e incluso llevar a cabo pruebas de integración y no solo unitarias.
* **Hecho:** Esta última columna está dedicada a colocar las secciones terminadas del proyecto.

Además, se implementó un mini tablero Kanban por cada miembro del equipo (en caso de que fuera necesario) al cual se le agregaron las subcolumnas de proceso y listo para el diseño, desarrollo y pruebas (QA) (ver ilustración 4). Para detallar más el proceso de desarrollo de la metodología se mostrará en un gráfico que contemple la creación de un único módulo con una o varias subtareas (ver tabla 2).



Ilustración 4. Mini tablero Kanban (gráfico propio).

Tabla 2. Pasos para llevar a cabo el desarrollo de elementos específicos (gráfico propio).

|  |  |  |
| --- | --- | --- |
| Paso | Elemento unitario | Elementos integrados o módulos |
| Verificar requerimiento | Cuando un requerimiento llaga se pone en la pila de trabajo y se abordan en ese orden, no debe pasar mucho tiempo para decidir si la tarjeta kanban se utilizará o no. | Se verifica que una funcionalidad pueda ser segmentada en más partes para resolver cada una de estas de forma particular. |
| Analizar el tiempo de desarrollo | De acuerdo con el nivel de prioridad que el CTO haga conocer se decide que requerimiento trabajar primero y cuáles son los elementos necesarios por conocer antes de comenzar a desarrollar. | Analizar el conjunto de requerimientos unitarios para estimar un tiempo de trabajo total sumando el tiempo de capacitación en caso de requerir una capacitación |
| Diseñar parte gráfica o lógica | El diseño gráfico incluye la selección de colores, formatos de imagen, iconos, distribución del contenido, medidas y selección de cada componente gráfico para la muestra de información. Si se trata de diseño lógico se hace referencia a la estructura del código, carpetas o métodos, el proceso de funcionamiento se detalla generalmente mediante un dibujo o gráfico para apreciar la idea. | Corroborar que el diseño total cuadre con los diseños específicos. |
| Desarrollar funcionalidad | Esta puede ser de tipo frontend o backend de acuerdo al tipo se trabajará con la capa del patrón de diseño que le corresponda. | Si el conjunto de un módulo desempeña sus funcionalidades individuales este apartado está completo |
| Probar elementos | Este apartado verifica que el requerimiento haya sido cumplido satisfactoriamente, de lo contrario se hace regresa al comienzo para revalorar la solución, se suele emplear la depuración de código. | La prueba de integración o de endpoint suele ser la más usada para verificar que un módulo efectivamente cumpla su función. |



# Métodos para cada objetivo específico

### Objetivo 1

### Objetivo 2

### Objetivo 3

### Objetivo 4

### Objetivo 5

# Viabilidad

Al aplicar el sistema web en la oficina de residencia se podría llevar acabo un mejor control de los registros en los alumnos, sus documentaciones y la liberación de sus formatos, con esto en cuenta se estima que en cuanto al proceso administrativo se ha tenido un progreso y reduce en parte la carga que tienen el personal del departamento de servicio social, abriendo una posibilidad de implementar un sistema similar a la residencia profesional o incluso en otras instituciones, sin embargo se debe tomar en cuenta que el sistema debe contar con un servidor dedicado para que no se sature o pueda caerse el sistema, ya que se estima que tenga una interacción considerable de usuarios, así mismo la información brindada en el sitio podría ser ignorada por el personal estudiantil provocando que sigan teniendo confusiones lo que causaría un efecto negativo mayor al actual.

Se mostrarán algunos puntos que permiten demostrar la viabilidad del proyecto basado en la experiencia de la empresa.

### Técnica

Al desarrollar este proyecto se hizo hincapié en algunos aspectos técnicos, como la organización e implementación de una estructura bien definida (conocida como de mvc) lo que se puede traducir como un trabajo que debería desarrollar unas mejores características de escalabilidad, pues el uso de este tipo de software le dan mayores oportunidades de crecimiento no solo dentro de un nivel operativo, sino que también dentro de una mejor calidad como profesionistas que conocen y saben manipular herramientas de desarrollo. Se visualiza un panorama favorable dentro del uso de este proyecto, pues sin duda era una necesidad que el software tenía que ser actualizado y no es la única vez que ha trabajado con este tipo de equipos de trabajo, por lo que la comunicación no se mostró como un problema.

### Económica

El uso de software empresarial a medida trae consigo una serie de gastos económicos mismos que pueden ser muy rentables dependiendo de la cantidad de demanda que se quiera cubrir, sin embargo con los sucesos del año 2020 se empezó a dar a conocer una nueva vertiente sobre la capacidad de llegar de forma remota hacia el público objetivo, es por ello que el desarrollo de software para empresas ayuda en gran medida a invertir los recursos económicos en herramientas que faciliten el trabajo a distancia, dado que el software que se implementó contiene código de versiones recientes, se estima que el momento en que se necesite de una actualización sea menor a 4 años dependiendo del crecimiento de la demanda de productos y que este tipo de actualización puede resultar bastante rentable en el futuro pudiendo llegar a durar hasta 10 años sin realizar una modificación de todo el código fuente representando un ahorro significativo para el instituto.

### Operativa

Debido a que los métodos que se solían utilizar en el departamento de Control Escolar dentro del Tecnológico, sería ampliamente recomendable realizar actualizaciones y mejoras dentro de las instalaciones, aunado a esto la capacitación para el uso del sistema web puede ser más productiva pues al no tener un espacio tan concurrido es más sencillo que los usuarios expresen sus inquietudes. No se observa un punto en el que el sistema deje de ser utilizado, sino por el contrario es muy probable que con el paso del tiempo se requiera mayores capacidades que agilicen la tarea visual de los administradores.



# Resultados

El sistema integral web para la gestión, control y seguimiento de residencias profesionales, servicio social y visitas a empresas, permite el control y seguimiento de trámites bajo los lineamientos del Tecnológico Nacional de México; para este sistema se implementaron cuatro módulos principales, que a su vez cuenta con submódulos y que se describen conforme a las iteraciones realizadas; como son:

* Jefatura de Control Escolar y vinculación: Administración de usuarios y catálogo de Empresas;
* Residencia profesional: documentos, evaluaciones, residentes, reportes, notificaciones, docentes y visitas;
* Servicio social: documentos, evaluaciones, residentes, reportes, notificaciones, docentes y visitas.
* Asesor Industrial: evaluaciones, documentos y liberación de constancia.
* Asesor Académico: evaluaciones, documentos, visitas, notificaciones.
* Jefatura de división: Administración de usuarios y catálogo de Empresas;

Cada uno de los módulos permite realizar altas, bajas, consultas y modificaciones e impresiones, según sea requiera. A continuación, se describen los módulos más importantes del sistema web:

**Creación de un módulo de inicio de sesión.**

Tener un control de los usuarios que ingresan al sistema es una parte fundamental, ya que con esto se puede mantener un control de acceso que ayuda a mantener un orden dentro de los procesos, con la creación de esta pantalla de login y sus validaciones se tiene cubierto el proceso de inicio de sesión, manteniendo un orden al hacer solo puedan iniciar sesión los usuarios que pertenezcan al instituto. (ilustración 4)

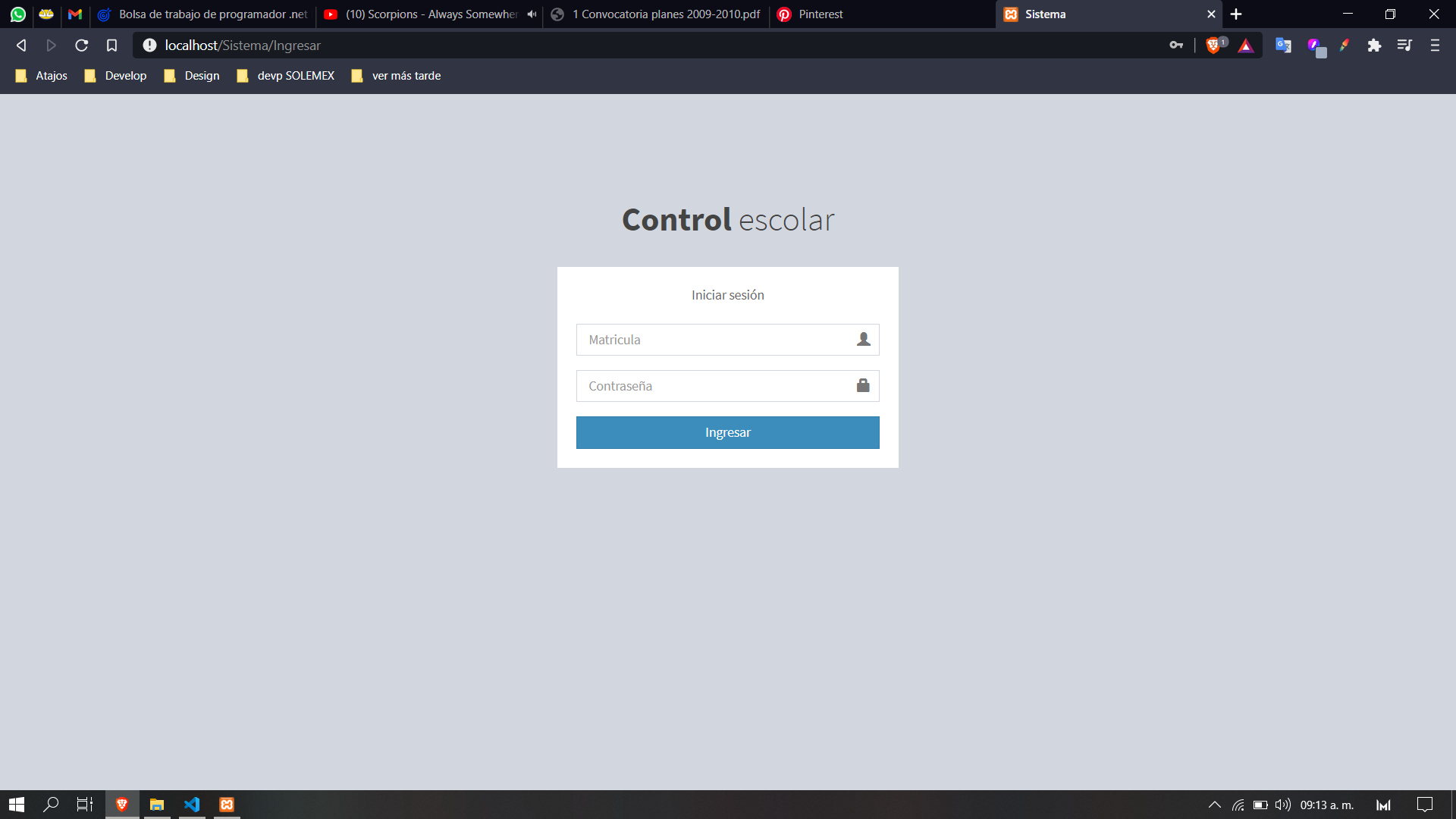


Ilustración 4 - Módulo de inicio de sesión.

**Establecer fechas para habilitar el proceso de registro**

Este módulo se encuentra solamente accesible por el administrador, por lo que el acceso se encuentra limitado por el rol de usuario, con ello se evita que cualquiera manipule los datos de registros. (ilustración 5)

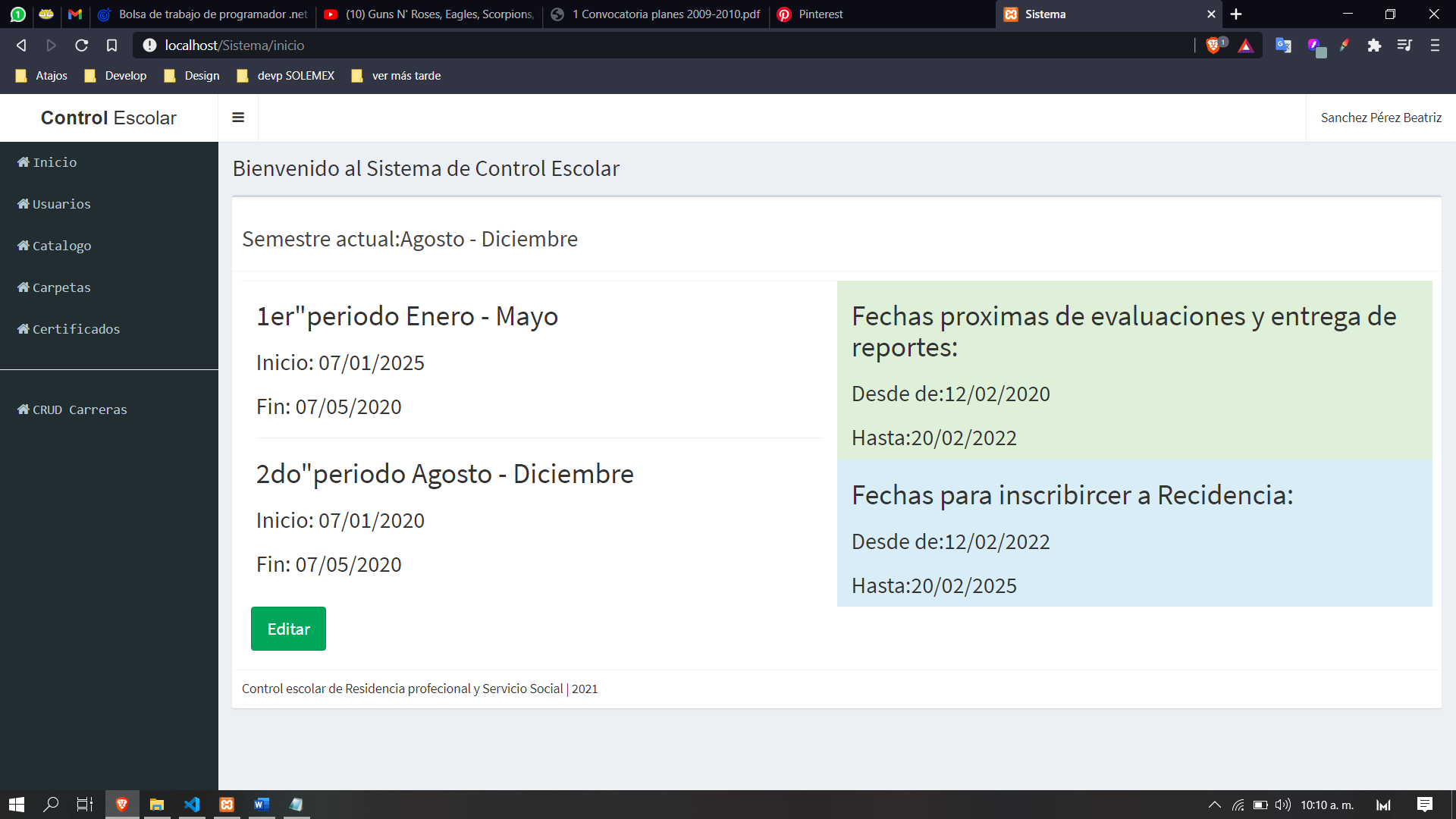


Ilustración 5 – Habilitar fechas y periodos.

En la siguiente imagen se muestra el módulo en el cual se podrá editar las fechas de registro y entrega reportes. Y fechas para inscripciones. (ilustración 6)

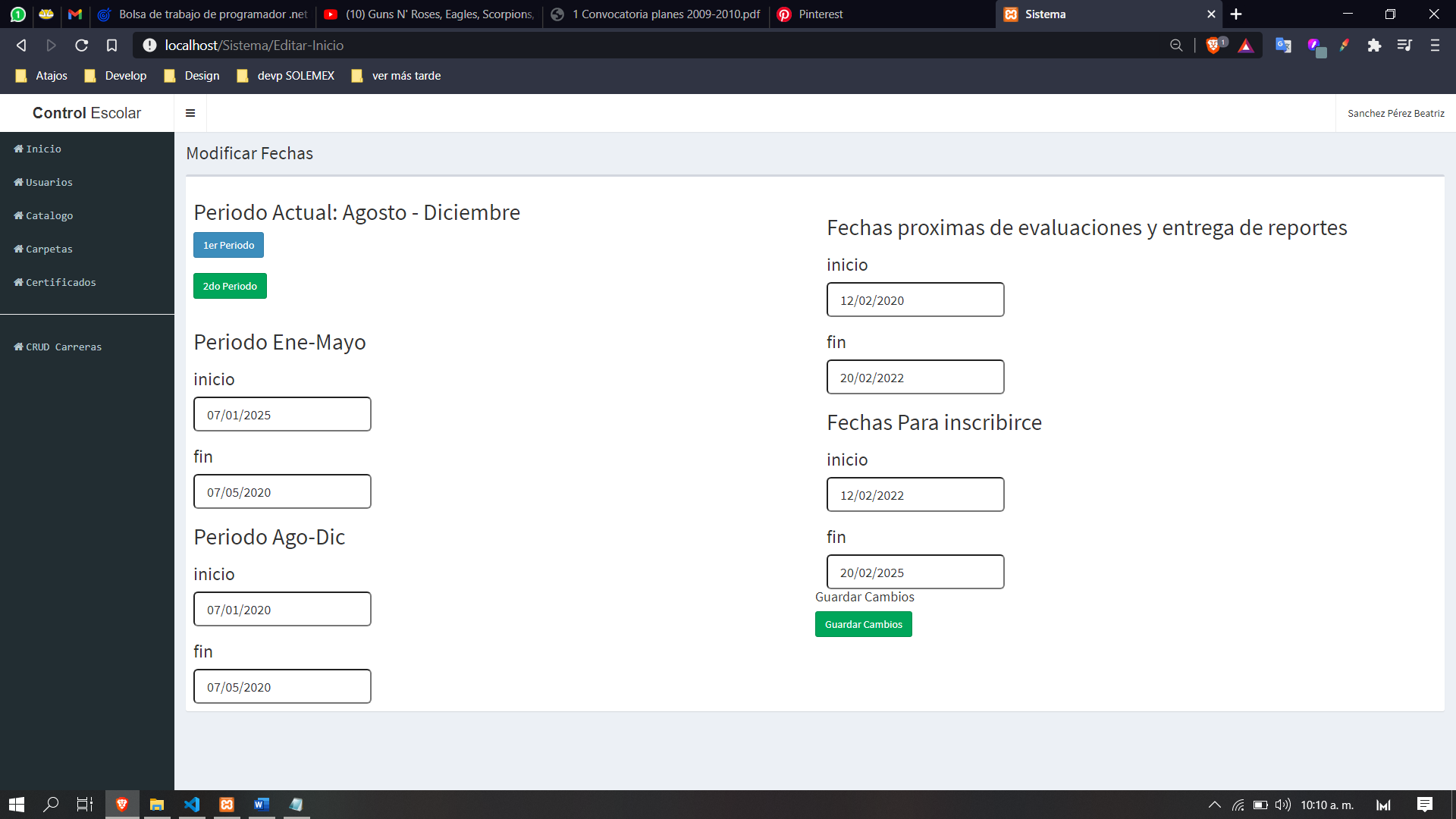
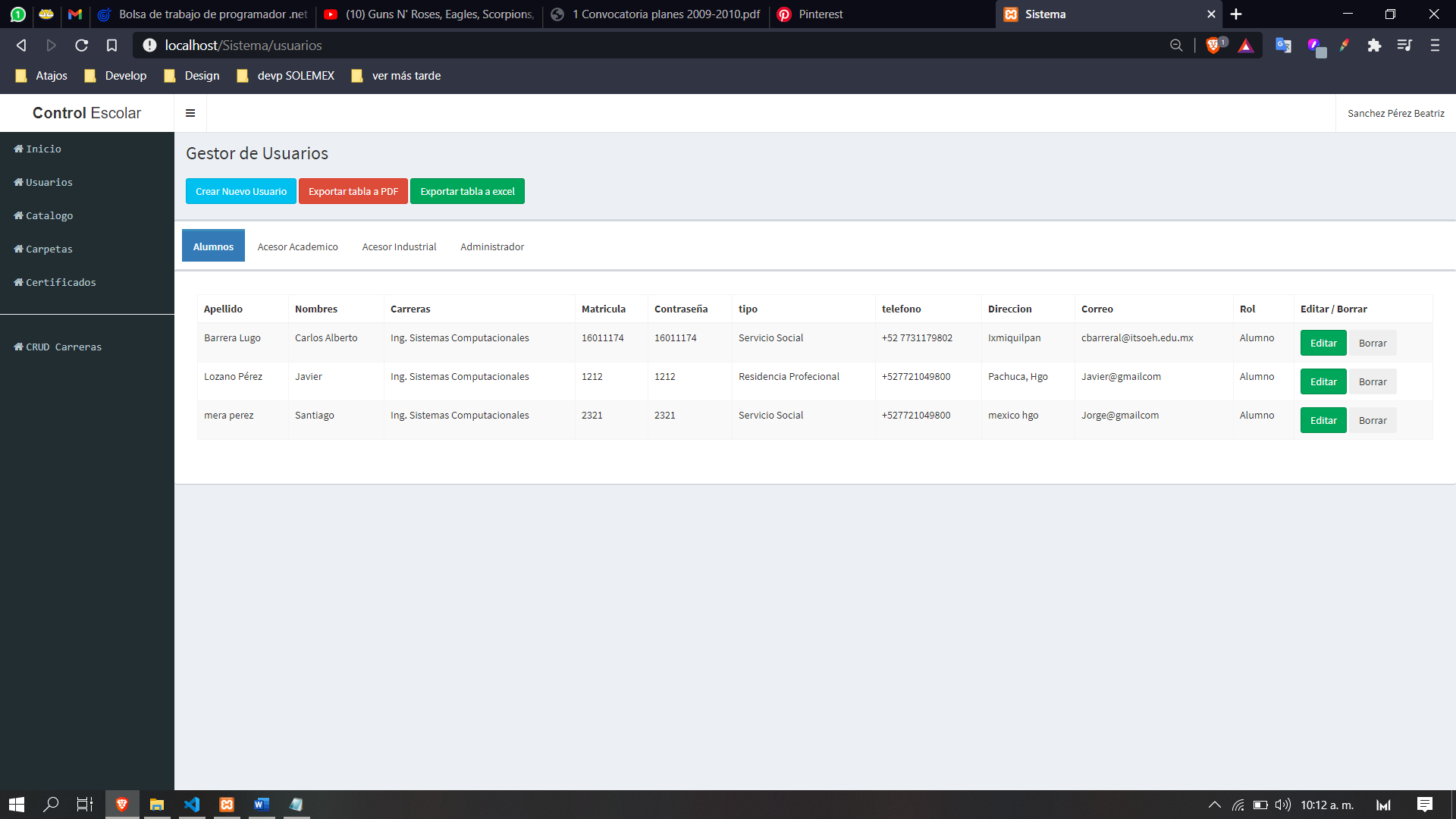


Ilustración 6 – Edición de fechas de evaluación y periodos.

**Usuarios**

En este módulo se permite el ingreso de los datos de los usuarios a la base de datos para posteriormente mostrarse en el sistema, con el fin de crear un mejor manejo de la información, (ilustración 7) a este módulo tienen acceso el usuario de tipo administrador, asesor académico, jefe de Carrera; Además permite el registro masivo mediante un documento Excel previamente llenado con la estructura correcta, asiendo el registro más rápido y más fácil (ilustración 8); Cuenta con la opción de exportar los datos de la tabla en un archivo Excel y en PDF. (ilustración 9 y 10)



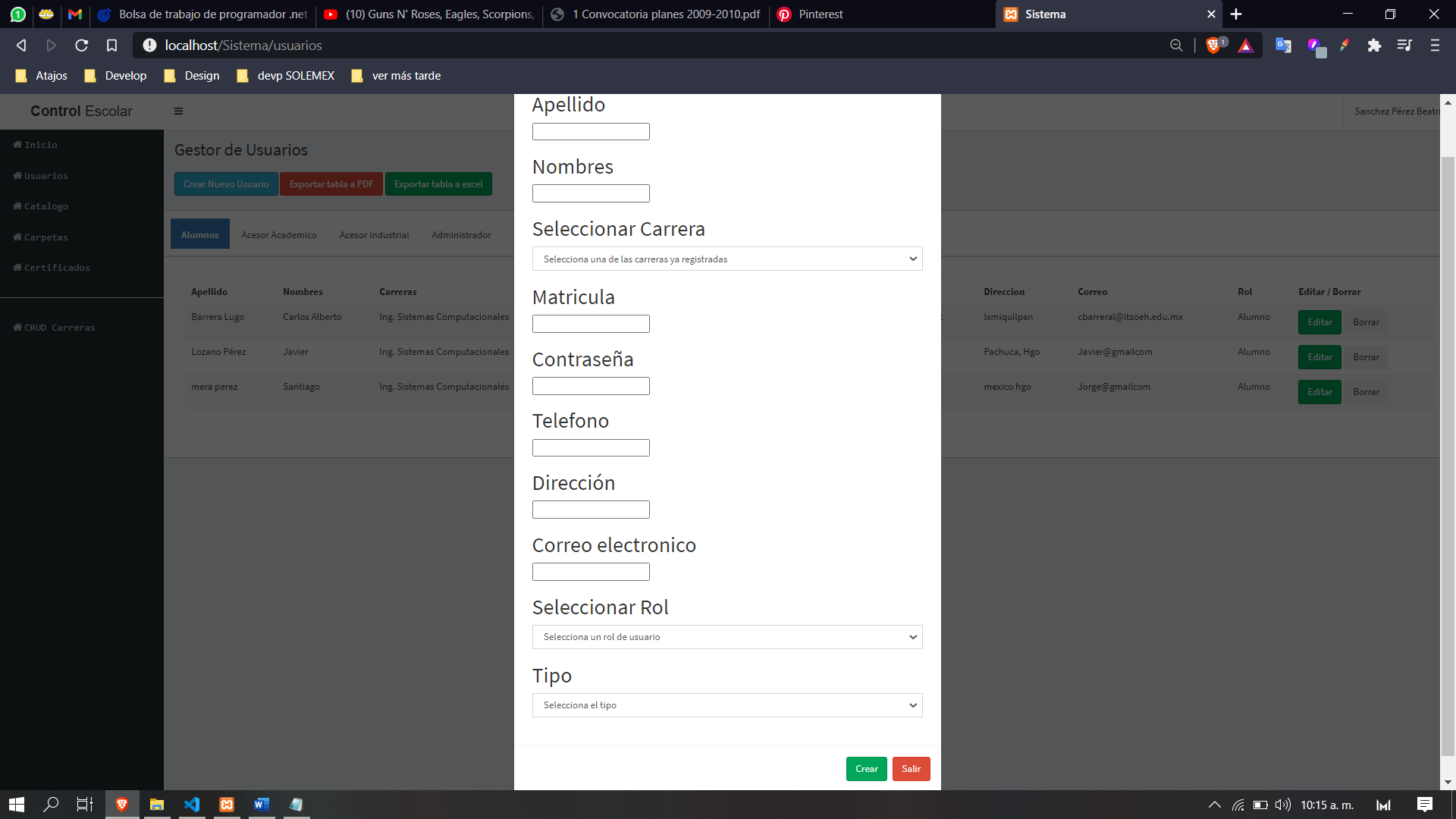


Ilustración 7 – Gestor de Usuarios

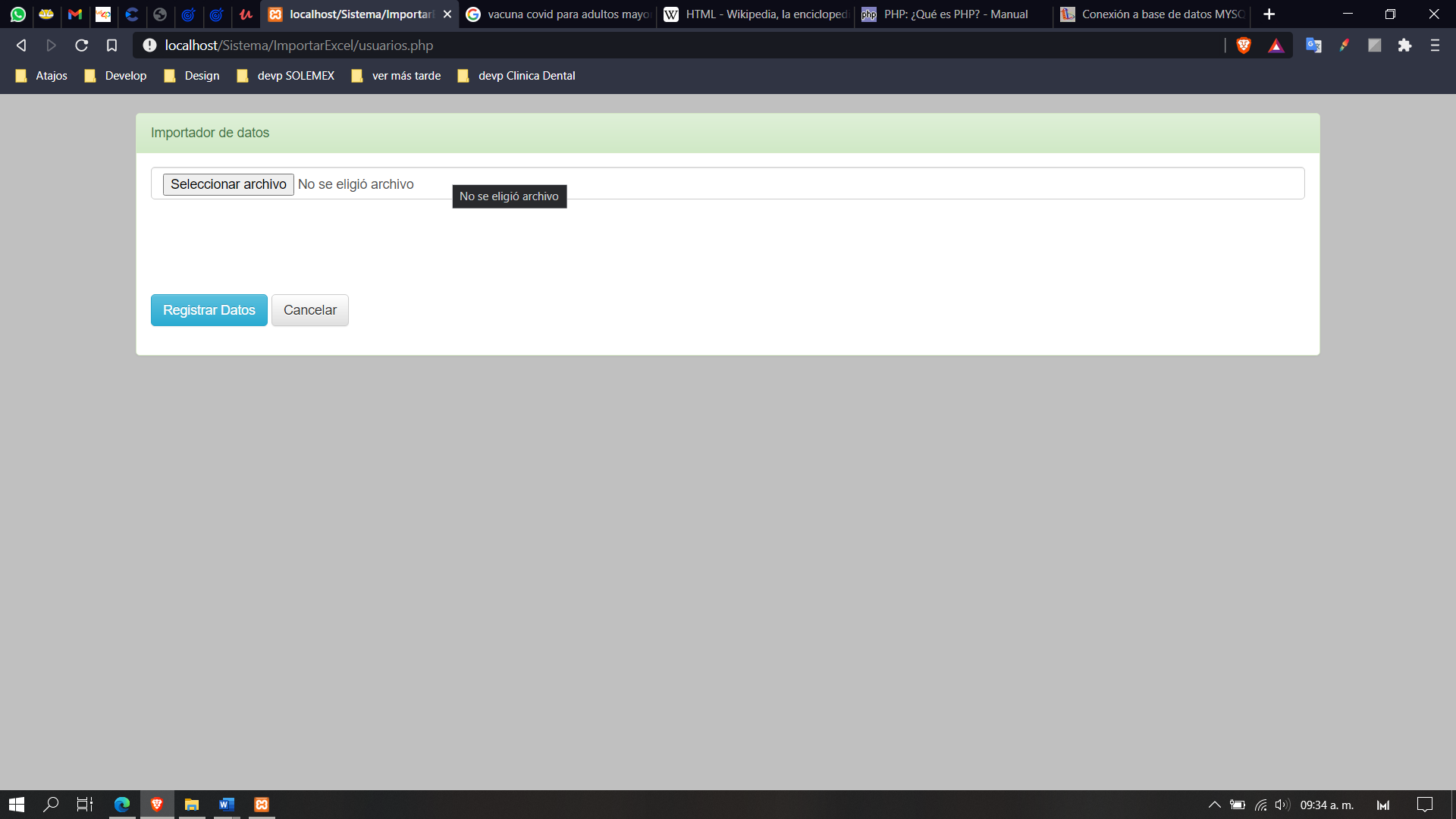


Ilustración 8 Importar datos desde Excel



Ilustración 9 Exportar tabla a PDF

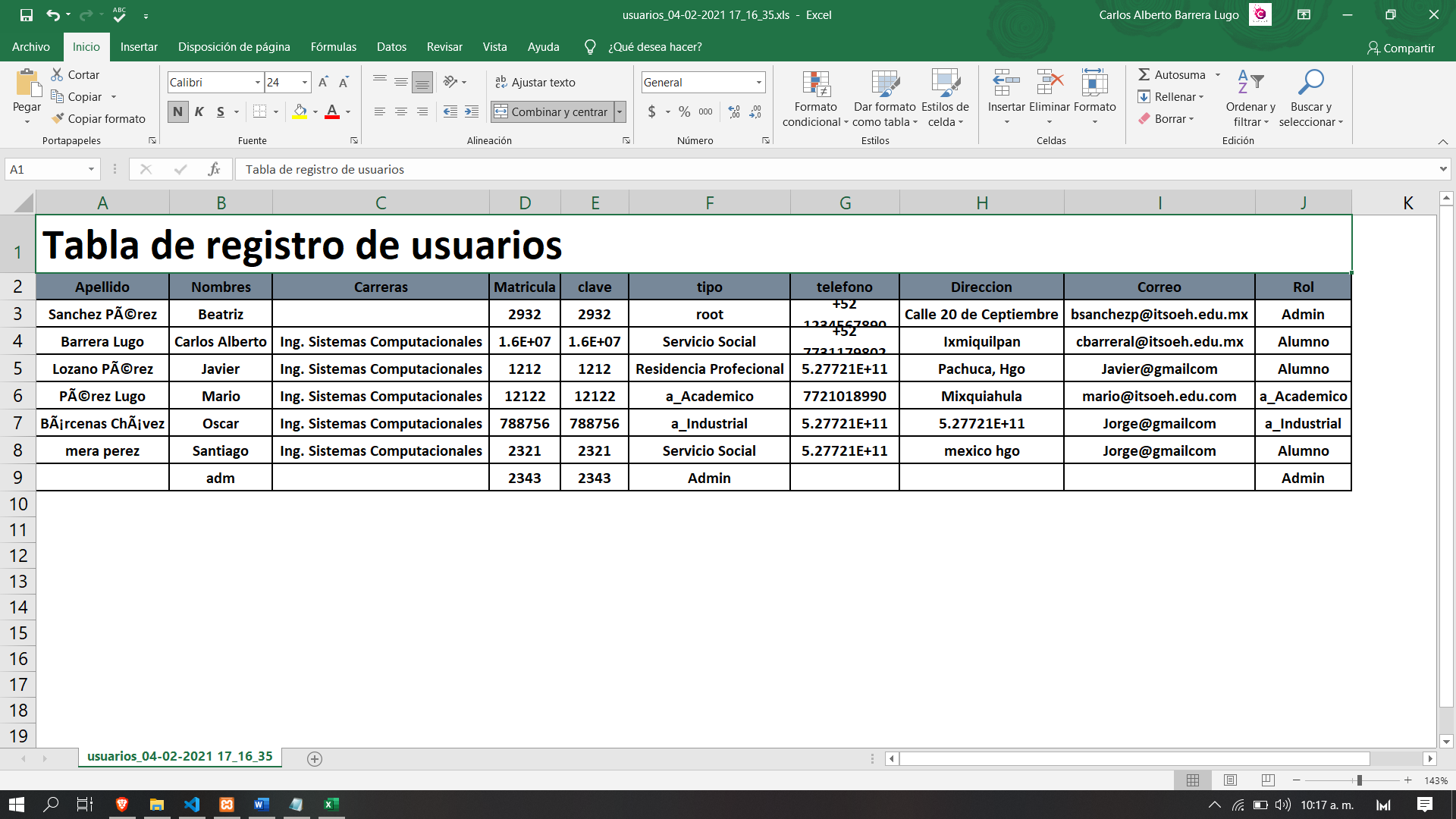


Ilustración 10 Exportar datos a Excel

**Módulo de Gestor de Programas educativos (carreras del Tecnológico)**

Este módulo solo puede ser accedido por el administrador, dentro de esta ventana permite habilitar/deshabilitar la opción de registro de los alumnos a empresas, cuenta con un buscador, permite vaciar la base de datos de la tabla carreras; Adjunto a esto, permite crear, editar y borrar los datos de las carreras y la visualización del catálogo de empresas y estudiantes que pertenecen a cada programa educativo (ilustración 11)

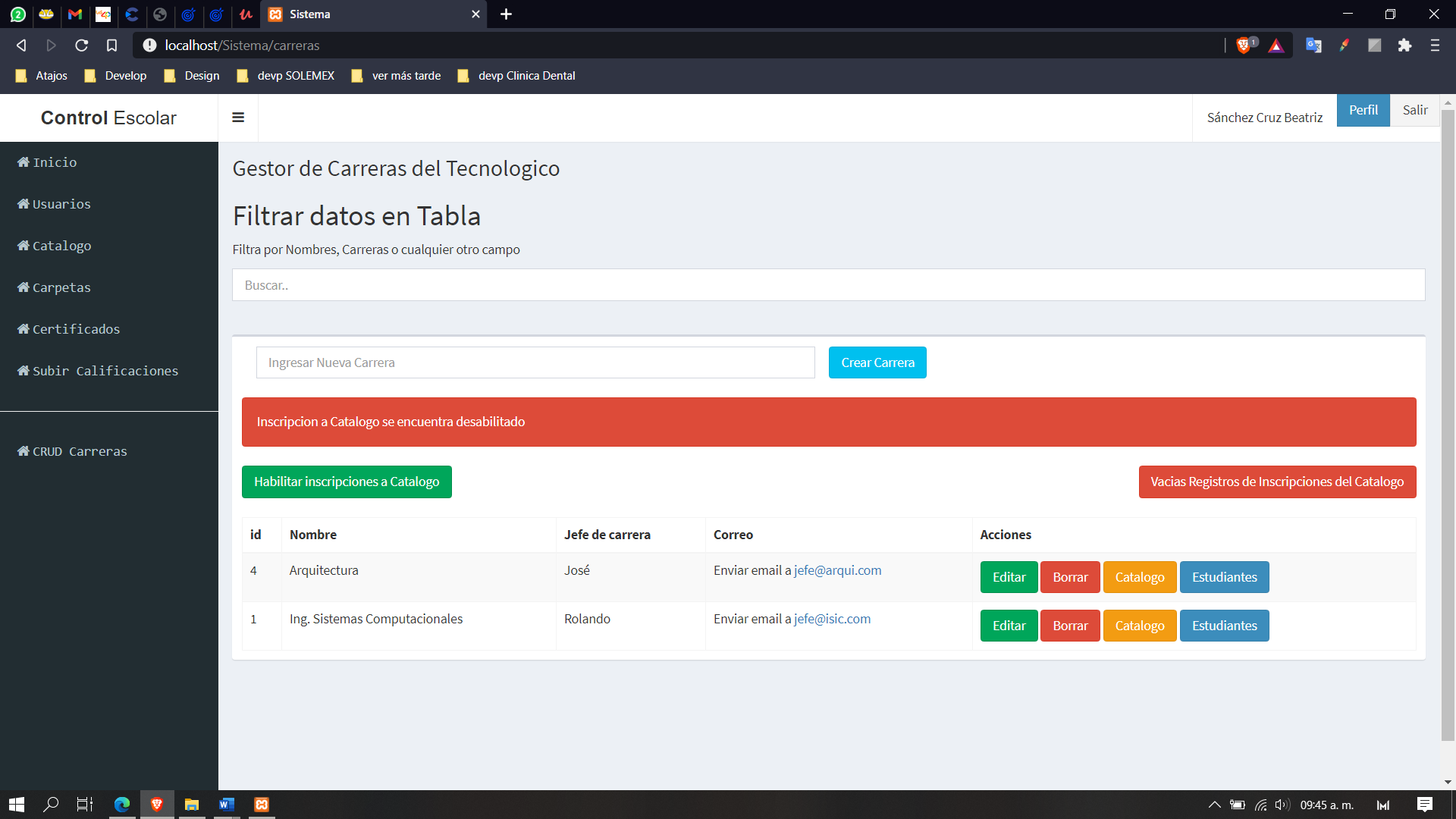
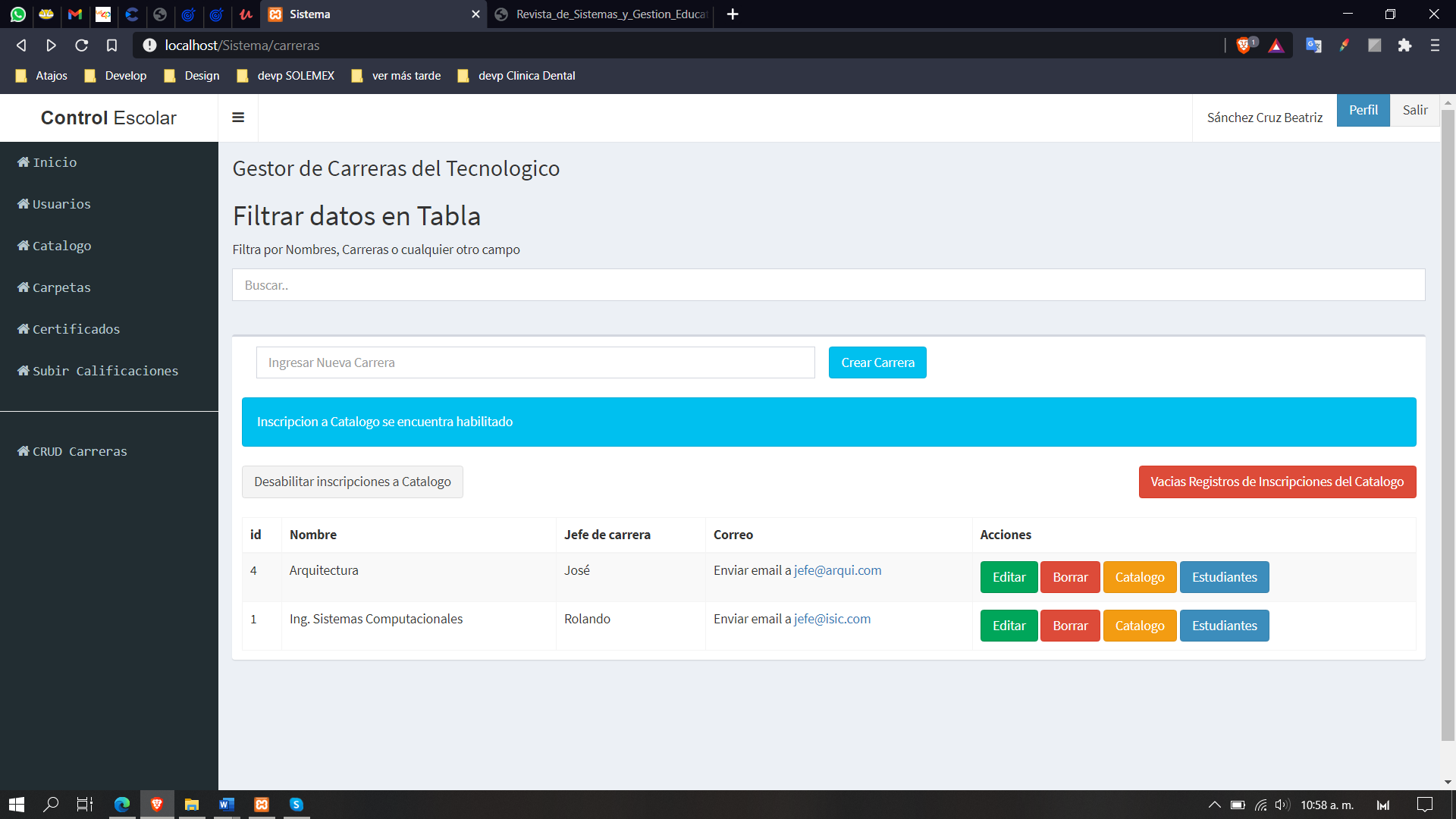
 

Ilustración 11 - Gestor de Carreras

**Módulo de Gestor de Catalogo de empresas**

Este módulo permite la gestión de las empresas asociadas al instituto, para el usuario tipo Alumno, solo puede visualizar el catalogo e inscribirse en alguna de las comisiones que tiene la empresa en el catálogo, (ilustración 12); así mismo también puede crear comisiones, es decir agregar horarios y cupo limite de alumnos que pueden inscribirse (ilustración 13); ambos módulos cuentan con la opción de exportación de datos a PDF

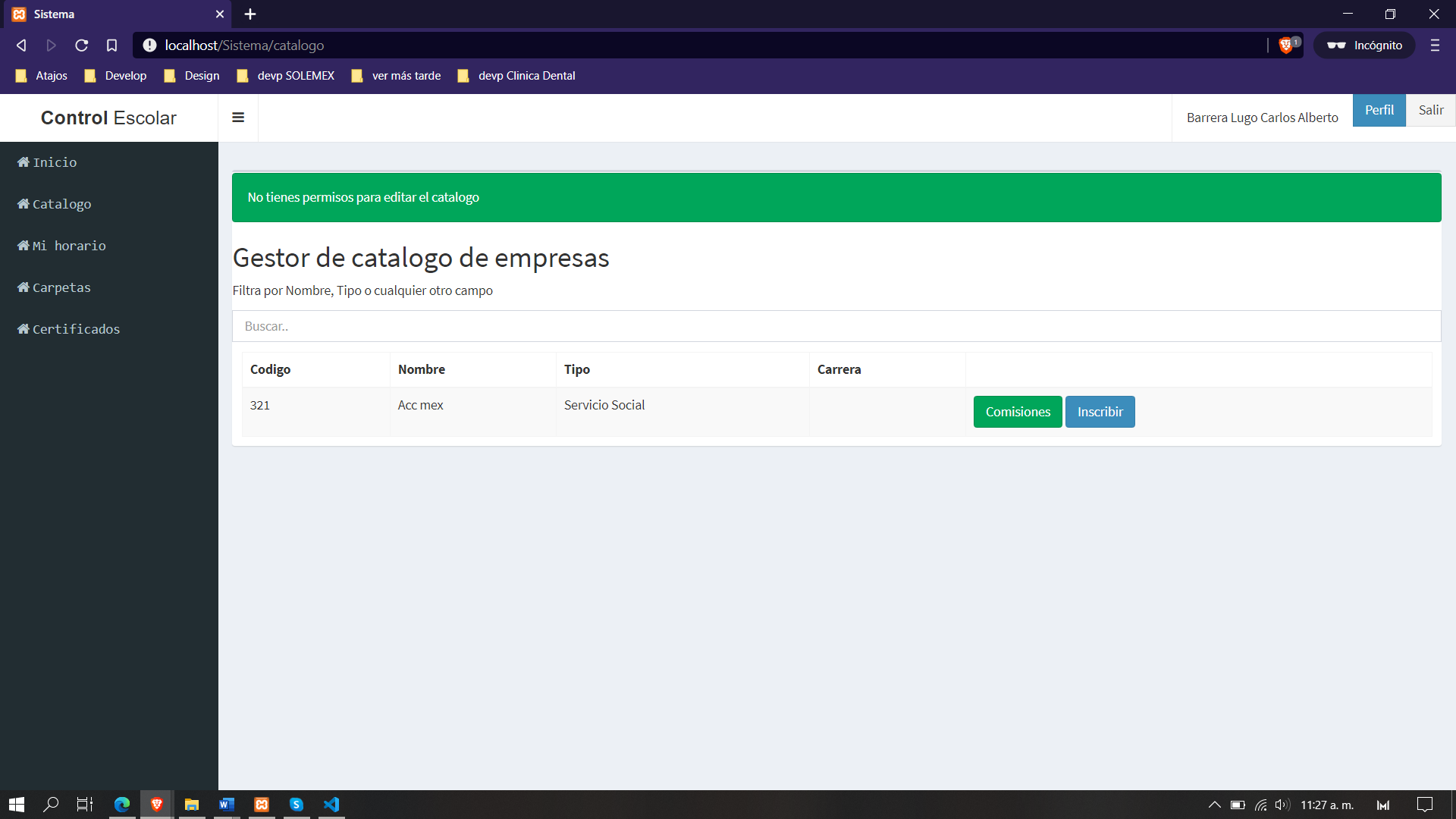
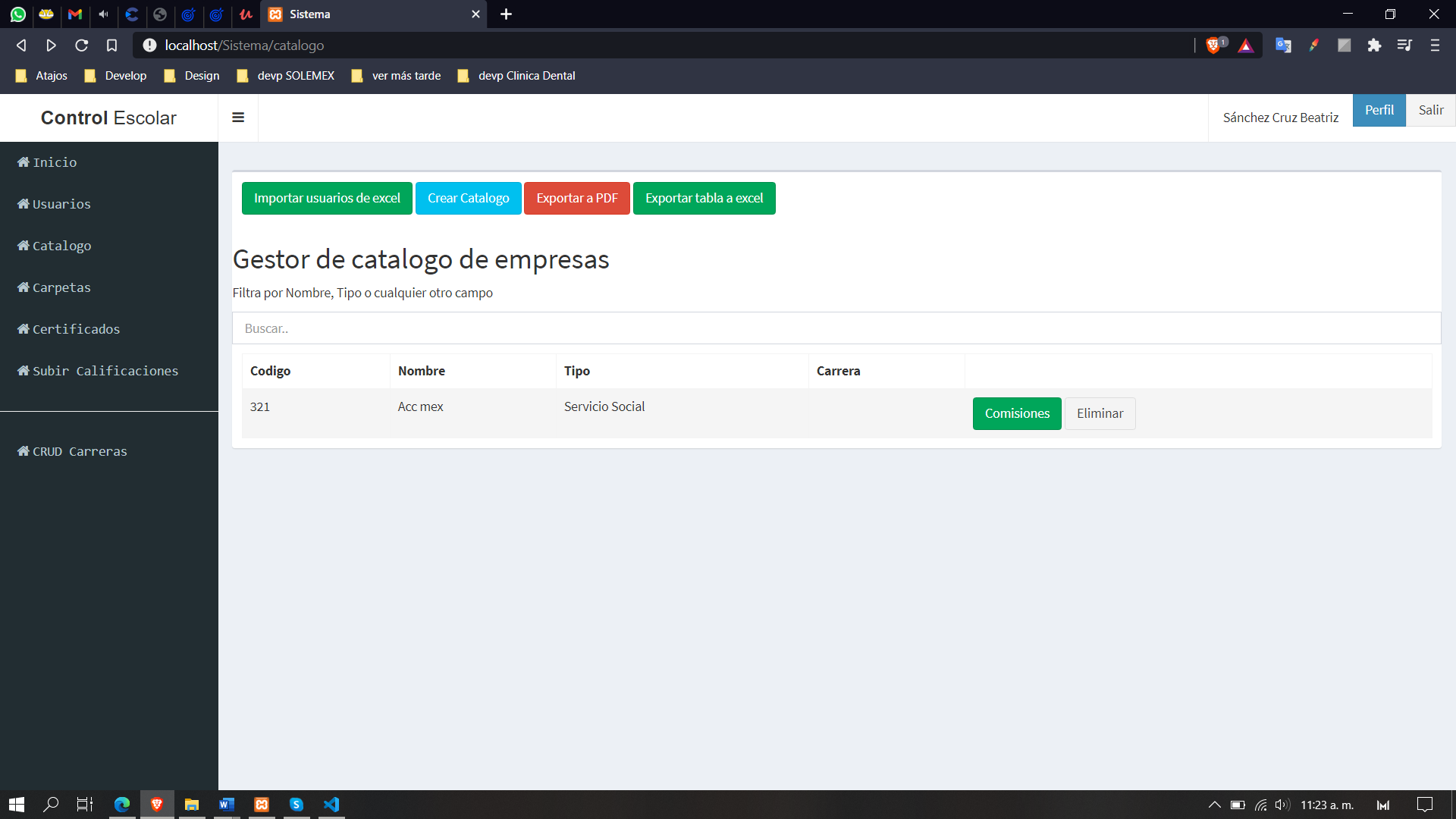
 

Ilustración 12 Catálogo de empresas

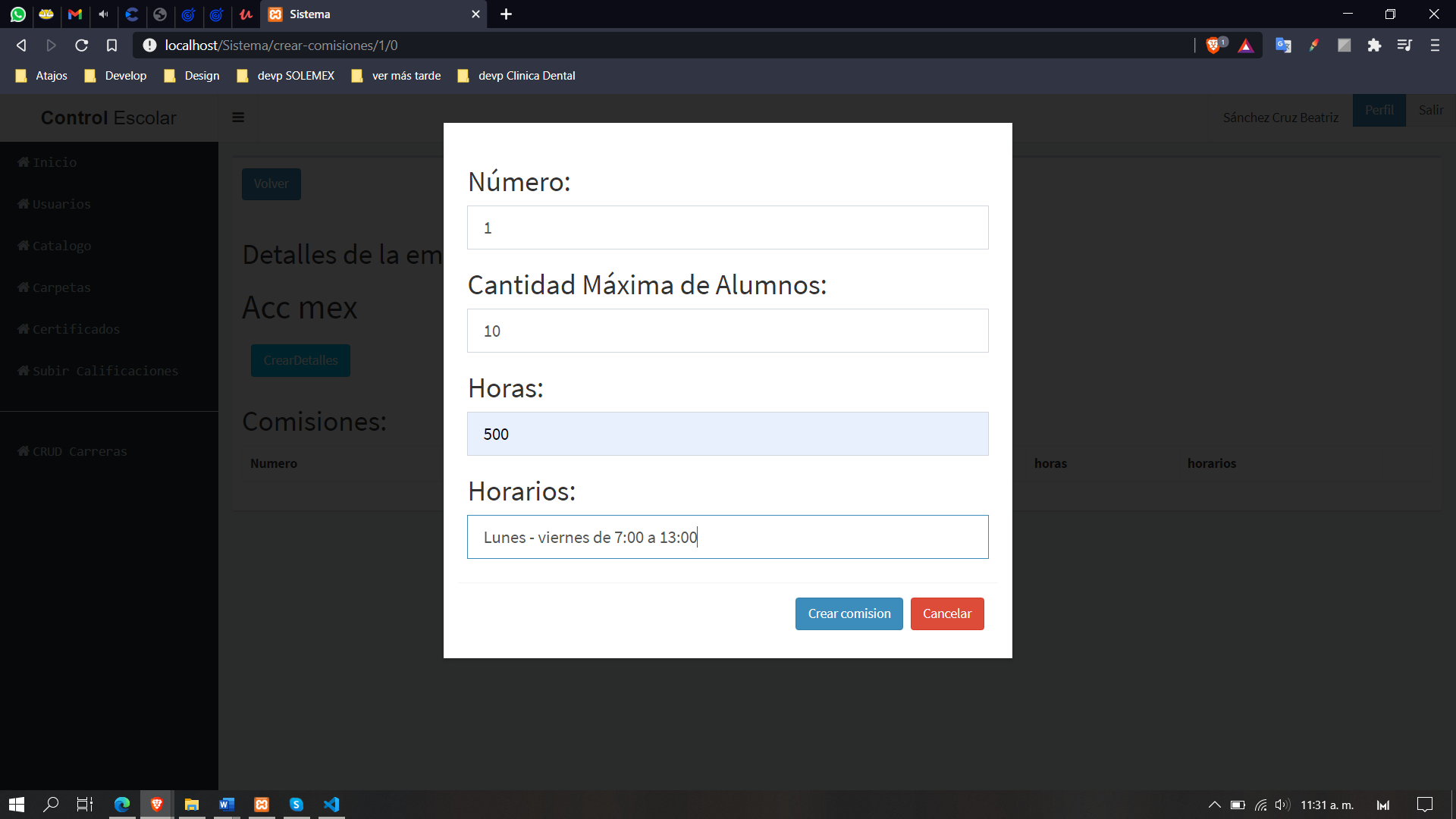
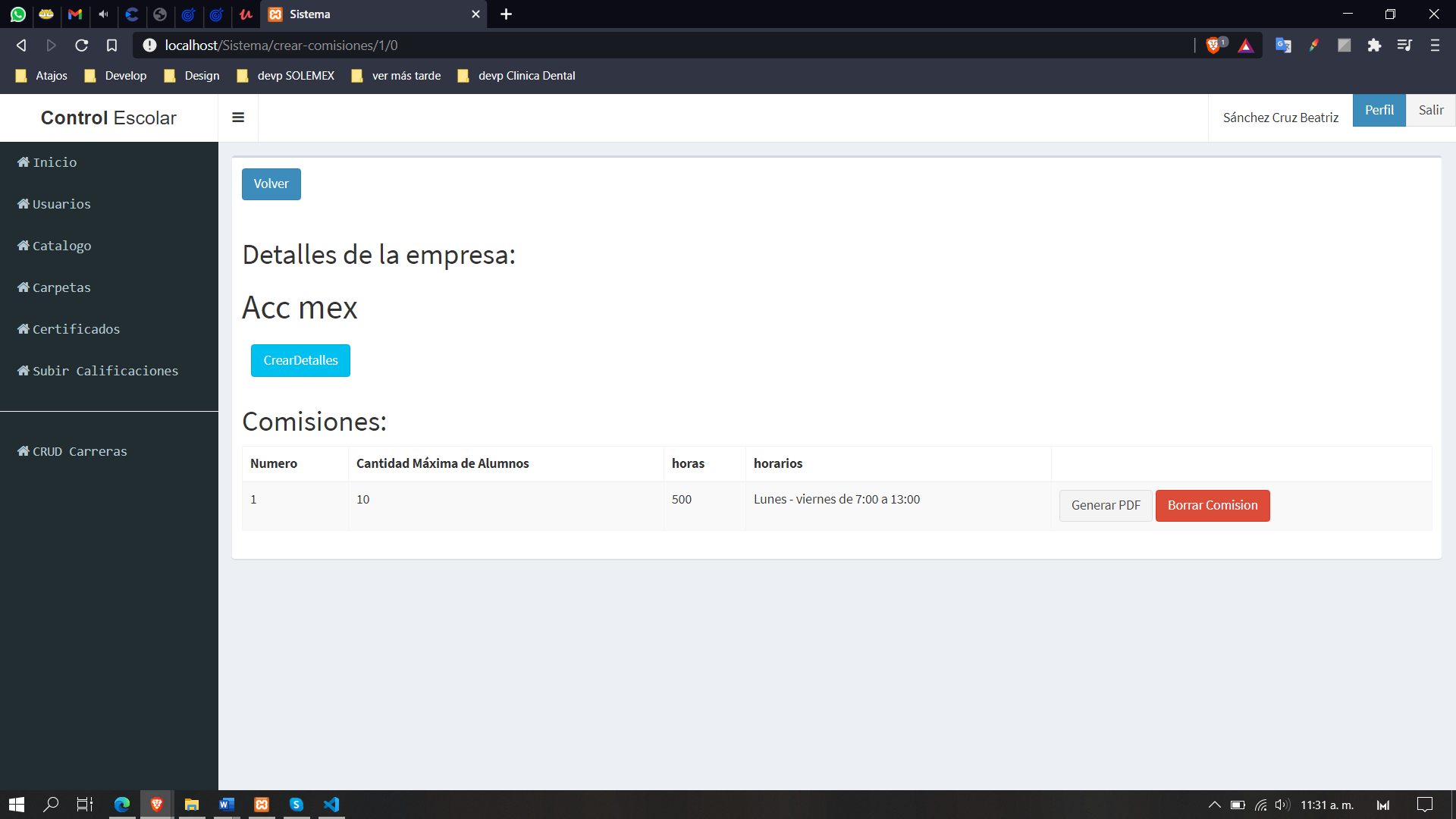
 

Ilustración 13 Creación y Visualización de Comisiones

**Módulo de Carpetas de alumnos**

Este módulo permite la visualización de alumnos tanto en Servicio social como residencia profesional, agiliza la búsqueda de un alumno con un buscador, además de que cuenta con un status de notificaciones, es decir cuando se realizan cambios en los documentos o evaluaciones, arroja una notificación de que la carpeta del alumno tiene modificaciones. (ilustración 14)

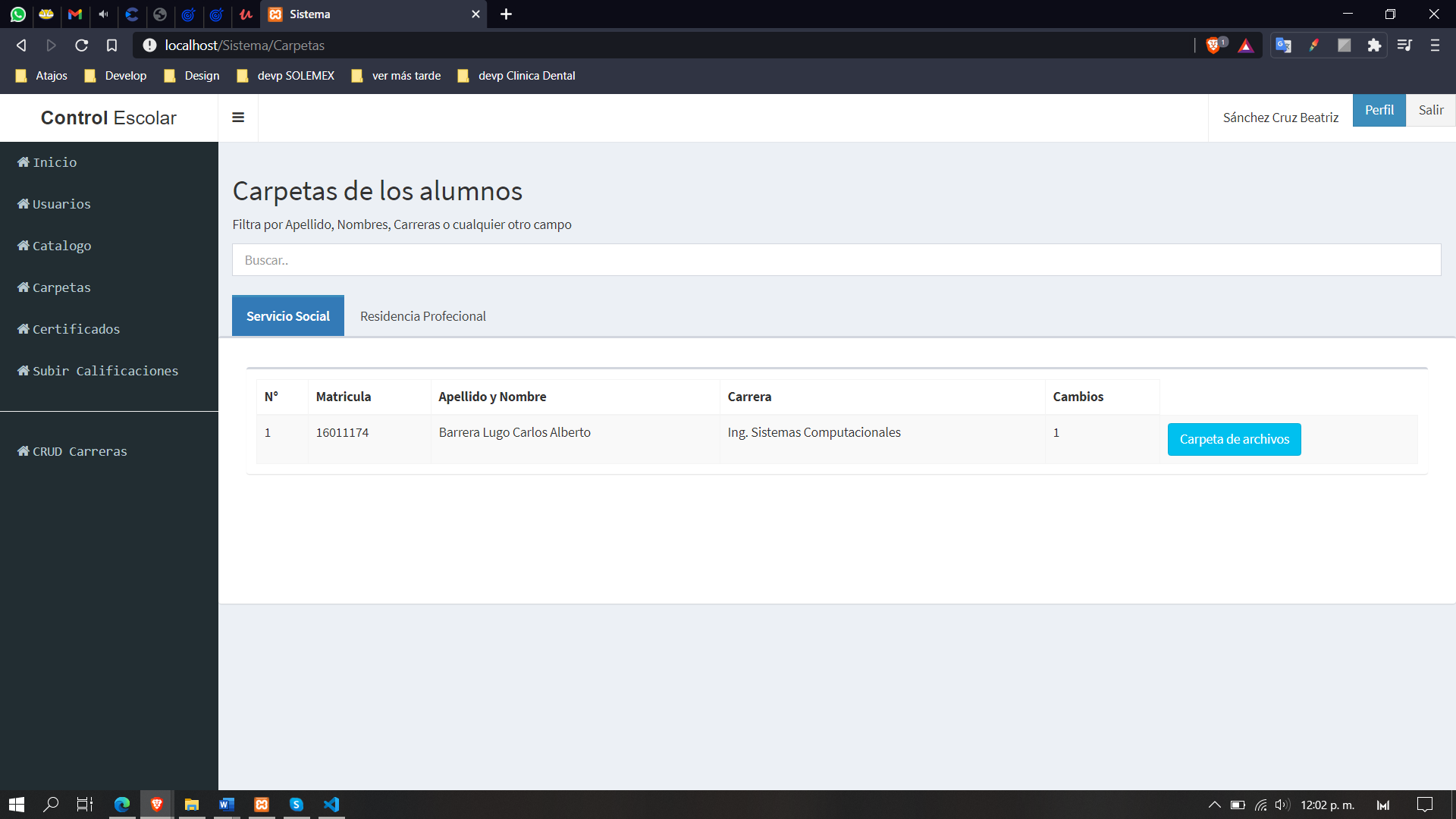


Ilustración 14 Carpetas de los alumnos

**Módulo Ver Carpetas de alumnos**

Este módulo permite que el usuario tipo Alumno gestione los documentos requeridos para el proceso de inscripción, registro, reportes y evaluaciones. (ilustración 15).

El usuario tipo Asesor Académico y Asesor Industrial pueden visualizar los documentos y a base de eso pueden evaluar el avance del alumno. (ilustración 16).

El usuario tipo jefe de Carrera, puede visualizar todo el contenido de la carpeta además de que después de revisarla notifica que ya ha sido aprobada; asiendo un filtro de seguridad que rectifica que el alumno está en orden. (ilustración 17).

*(Ilustraciones de la 15, 16, 17 pendientes)*

**Módulo Evaluaciones**

Este módulo permite crear y editar la evaluación por reporte del alumno, dicho apartado tiene acceso el asesor académico y el asesor industrial, quienes son los que evalúan el avance de la estadía (ilustración 18).

*(Ilustración 18 pendiente)*

**Módulo Subir evaluaciones**

Este módulo filtra los usuarios que cumplen con ambas evaluaciones de los asesores, y con el estatus de revisado del jefe de carrera, cuenta con un sistema de notificación, ya que muestra una tabla con el nombre del alumno, la matricula, un promedio final del reporte subido, así mismo también un botón de marcar como “hecho” y otro botón que redirecciona a la carpeta del alumno para rectificar que ya fue revisado y está listo (ilustración 19).

Este módulo tiene acceso el Asesor Académico, el jefe de Carrera y el Administrador

*(Ilustración 19 pendiente)*

**Módulo de solicitar constancia**

Este módulo se activa automáticamente al finalizar el periodo y siempre cuando el alumno cumple con todas las evaluaciones y reportes satisfactoriamente, se habilita un botón para hacer la petición de la constancia de liberación, esta solicitud llega al asesor industrial (ilustración 20), una vez que el asesor industrial genera la constancia en formato PDF, se solicita la constancia al administrador (Jefatura de Control escolar). (ilustración 21)

*(Ilustraciones 20, 21 están pendientes)*

**Módulo de notificaciones mediante Correo electrónico**

Este módulo envía un Gmail al usuario, trabaja de forma interna, hace saber de forma automática al detectar cualquier movimiento del usuario, tanto como fechas de entrega de reportes, fechas de evaluación, cuando inicia y termina el periodo, notifica si ya fue evaluado el alumno. (ilustración 22).

*(Ilustración 22, está pendiente)*



# Conclusiones

El uso de las tecnologías web para el desarrollo e implementación del sistema integral web para la gestión, control y seguimiento de residencias profesionales, servicio social y visitas a empresas trae para el personal del departamento de gestión tecnológica y vinculación, docentes y estudiantes del Instituto Tecnológico de Oaxaca la disposición de la información estructurada y organizada en cualquier momento y desde cualquier lugar con una conexión a internet. Con el proyecto se beneficia en gran medida a los estudiantes residentes y prestadores del servicio social, docentes, ya que cuenta con un control eficiente del expediente de cada estudiante que se encuentre realizando la residencia profesional; cada residente conoce de manera oportuna en qué parte del trámite se encuentra, ya que el sistema lleva un seguimiento de cada proceso, y es notificado sobre tiempos de entrega de documentos para evitar confusión, bajas por incumplimiento y tiempos de espera prolongados.

# Recomendaciones

Dado a lo prematuro del proyecto este aún puede mejorase en el aspecto de los módulos en los usuarios, así mismo se pueden optimizar las acciones de adición de usuarios y el manejo de los permisos, una vez optimizado esos procesos se puede ver la posibilidad de ampliar el sistema a titulación y/o implementarse en otros institutos.

# Bibliografía

Anderson, D. J., & Carmichel, A. (2016). *Essential Kanban Condensed.* Seattle: LEAN Kanban University Press.

Arnold Cathalifaud, M., & Osorio, F. (1998). Introducción a los Conceptos Básicos de la Teoría General de Sistemas. *Cinta de Moebio*(3). Obtenido de https://www.redalyc.org/pdf/101/10100306.pdf

Camps Paré, R., Casillas Santillán, L., Costal Costa, D., Gibert Ginesta, M., Martín Escofet, C., & Pérez Mora, O. (2005). *Bases de datos.* Barcelona: UOC.

Dooley, J. F. (2017). *Software development, design and coding. with patterns, debugging, unit testing and refactoring.* Galesburg, Illinois: Apress.

Dykes, L., & Tittel, E. (2005). *XML for Dummies.* Hoboken: Wiley Publishing.

Escudero Cisneros, J. (2017). *Diseño de sistema de control y gestión de proveedores y agentes para inmobiliaria.* Obtenido de Universidad Autónoma del Estado de Hidalgo: https://www.uaeh.edu.mx/docencia/Tesis/huejutla/licenciatura/

Fernandes, C. (s.f.). *Styde*.

Git. (2014). *Documentation: 1.1 Inicio - Sobre el control de versiones*. Obtenido de Git: https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones

Gómez Fuentes, M. d. (2013). *Notas del curso BASES DE DATOS.* CDMX: Casa abierta al tiempo UAM Cuajimalpa.

Gonzalez Godino, F. (2013). *Diseño de una arquitectura escalable y de alta disponibilidad para un sistema middleware.* Obtenido de Universidad Politécnica de Madrid: http://oa.upm.es/32647/1/PFC\_FERNANDO\_GODINO\_GONZALEZ.pdf

Gorman, K., Hirt, A., Noderer, D., Rowland-Jones, J., Sirpal, A., Ryan, D., & Woody, B. (2019). *Introducing Microsoft SQL Server 2019. Reliability, scalability, and security both on premises and in the cloud.* Birmingham: Packt Publishing.

Johansen Bertoglio, O. (1993). *Introducción general a la teoría general de sistemas.* Santiago: LIMUSA GRUPO NORIEGA EDITORES.

Landa Cosio, N. (25 de Febrero de 2018). LINQ C# en español, básico. *Este es un curso básico de LINQ con C#, nos enfocamos principalmente en LINQ para objetos y XML.* Obtenido de https://www.youtube.com/playlist?list=PLM-p96nOrGcYGyfYpStN3jnaTuIosO3V5

Landa Cosio, N. A. (2018). Tutorial XML en español. *Curso básico de XML para entenderlo de forma rápida y sencilla*. Obtenido de https://www.youtube.com/playlist?list=PLM-p96nOrGcYb96AMy3VdUN8fo-sVAx7K

López Menéndez de Jiménez, R. E. (Diciembre de 2015). Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales. *ITCA-FEPADE*, 6-11. Obtenido de https://core.ac.uk/download/pdf/80296686.pdf

Luna López, C. A. (2012). *Estructura de datos.* Estado de México: Red tercer milenio.

Merizalde Andrade, C. E., & Toledo Vallejo, A. F. (2013). *Repositorio Institutcional de la Universidad Politécnica Salesiana.* Obtenido de Desarrollo e implementación de un sistema de escritorio para el control de producción: https://dspace.ups.edu.ec/bitstream/123456789/4229/1/UPS-GT000382.pdf

Mompó Camarasa, Á. (2017). *Repositorio Institucional Universitat Politècnica de València.* Obtenido de Estudio de sistema de control de versiones y uso de los mismos en proyectos de integración continua: https://riunet.upv.es/bitstream/handle/10251/87104/MOMP%C3%93%20-%20Estudio%20de%20sistema%20de%20control%20de%20versiones%20y%20uso%20de%20los%20mismos%20en%20proyectos%20de%20integraci%C3%B3....pdf?sequence=1

Morero, F. (2000). *Introducción a la OOP.* Obtenido de Laboratorio de computación de la universidad Magallanes: https://kataix.umag.cl/~ruribe/Utilidades/Introduccion%20a%20la%20Programacion%20Orientada%20a%20Objetos.pdf

Mozilla and contribuitors. (23 de Marzo de 2019). *CRUD*. Obtenido de MDN web docs: https://developer.mozilla.org/es/docs/Glossary/CRUD

Pérez Filoche, Á. (Enero de 2020). *Archivo Digital Universidad Politecnica de Madrid.* Obtenido de Evoluciones y refactorizaciones de una aplicación Android para alfabetización digital: http://oa.upm.es/58085/1/TFG\_ALVARO\_PEREZ\_FILOCHE.pdf

Pítsica Marques, M. (2001). *SISTEMA DE INFORMACIÓN PARA LA GESTIÓN APLICADO EN LAS ENTIDADES FINANCIERAS: ESTUDIO EMPIRICO SANTA CATARINA.* Obtenido de Universidad Complutense Madrid: http://webs.ucm.es/BUCM/tesis//cee/ucm-t25237.pdf

Pressman, R. (2010). *Ingeniería del software. Un enfoque práctico.* Ciudad de México: McGrawll Hill.

Project Management Institute. (2017). *AGILE PRACTICE GUIDE.* Pennsylvania: PMI Book Service.

Red Hat Inc. (2020). *El concepto de IDE*. Obtenido de MiddleWare: https://www.redhat.com/es/topics/middleware/what-is-ide

Refactoring Guru. (2014-2020). *Patrones de diseño*. Obtenido de Refactoring Guru: https://refactoring.guru/es/design-patterns/what-is-pattern

Richards, M. (2015). *Software architecture patterns.* Sebastopol, CA: O’Reilly Media, Inc.

SAP. (2020). *SAP History*. Obtenido de SAP: A 48-year history of success: https://www.sap.com/corporate/en/company/history.html

Sebastián Gómez, L. (2000). *Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación.* Obtenido de ResearchGate: https://www.researchgate.net/publication/228877430\_Diseno\_de\_Interfaces\_de\_Usuario\_Principios\_Prototipos\_y\_Heuristicas\_para\_Evaluacion

Stack Overflow Comunity. (07 de Marzo de 2017). *C# Notes for professionals*. Obtenido de GoalKicker.com: https://goalkicker.com/CSharpBook/

Suárez Rey, C. (2010). *Sistemas integrados de gestión.* Obtenido de Grupo Jean Monet de Competencia y desarrollo regional: http://www.gcd.udc.es/subido/catedra/presentaciones/economia\_competencia\_ii/nota\_tecnica\_sistemas\_de\_gestion\_erp\_carlos\_suarez\_rey\_17-03-2010.pdf

Suryanarayana, G., Samarthyam, G., & Sharma, T. (2015). *Refactoring for software design smells. Managing technical debt.* Waltham, MA: Elsevier.

The Institute of Electrical and Electronics Engineers. (28 de Septiembre de 1990). IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*. New York, New York, United States of America: IEEE Standards.

Yépez Llerena, E. D., & Armijos Guillen, K. F. (2020). *Repositorio de la UNACH.* Obtenido de APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE CONTROL ACADÉMICO UNACH.: http://dspace.unach.edu.ec/bitstream/51000/6457/1/APLICACI%C3%93N%20DE%20LA%20METODOLOG%C3%8DA%20KANBAN%20EN%20EL%20DESARROLLO%20DEL%20SOFTWARE%20PARA%20GENERACI%C3%93N.pdf

Expresiones Regulares PHP, Diego Larazo, (2020), *Repositorio de la UNAM.* https://diego.com.es/expresiones-regulares-en-php

De miguel Adoración., Piattini Mario., Marcos Esperanza. (2000). Diseño de base de datos relacionales. Alfaomega

Eguíluz Pérez Javier. (Junio de 2008). Introducción a JavaScript. Recuperado de https://goo.gl/y0cWEG

Gómez, I., Pegenaute, X., Reig, G., & Torres, J. (2009). Cloud Computing una visión desde un aula de la UPC. Sevilla, España: Lulú.

Heurtel, O. (2014). PHP y MySQL: domine el desarrollo de un sitio web dinámico e interactivo. Ediciones ENI.

Kendall, J. E. (2005). Análisis y diseño de sistemas. Pearson Educación. Lafosse, J. (s.f.). Struts 2: El framework de desarrolo de aplicaciones Java EE 2014. Cornellà de Llobregat (Barcelona): ENI.

Laudon, K., & Laudon, J. (2004). Sistemas de información gerencial (Octava ed.). México: Pearson

Lineamientos para la Operación y Acreditación de la Residencia Profesional versión 1.0 Planes de Estudio 2009-2010, Fecha de creación 27 de octubre de 2011

Lineamiento para la Operación y Acreditación del Servicio Social versión 1.0 Planes de Estudio 2009- 2010, Fecha de creación 2 de septiembre de 2011.

Luján Mora Sergio. Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario, Alicante, 2002. ISBN: 978-84-8454-206-3. (PI'2002) Libro / Book

Nevado, M. (2010). Introducción a Las Bases de Datos Relacionales. España: Visión Libros

Pressman, R., & Murrieta, J. (2006). Ingeniería del software un enfoque práctico (6ª Edición ed.). McGraw-Hill.

Sommerville, Ian M. I. (2005). Ingeniería del software. Pearson Educación.

Zaefferer Jörn. (s.f). Validación de formularios con jQuery. Recuperado de https://Jqueryvalidation.org/

Sergio Luján Mora (2001). Programación en Internet: Clientes Web (libro completo gratuito en pdf) (1ª edición). Editorial Club Universitario.

Sergio Luján Mora (2002). Programación de aplicaciones web: historia, principios básicos y clientes web (libro completo gratuito en pdf) (1ª edición). Editorial Club Universitario.

# Anexos

## Manual de usuario

*(Manual de usuario, PENDIENTE)*