February 28th, 2025 at 21:32 UTC



**Plagiarism Detection Report** 

JustDone Al

## **Plagiarism Detection**

0.2%

Plagiarism Types	Text Coverage	Words
Identical	0%	0
Minor Changes	0.2%	24
Excluded		
Omitted Words		0

Total Pages:

60

Total Words:

14869









# **Plagiarism**

0.2%

#### Results (1)

Repository	Internal Database
N/A	0
	<b>→</b>
Internet Sources	Current Batch
1	0

Plagiarism Types	Text Coverage	Words
Identical	0%	0
Minor Changes	0.2%	24
Excluded		
Omitted Words		0

#### About Plagiarism Detection

Our Al-powered plagiarism scans offer three layers of text similarity detection: Identical, Minor Changes, and Paraphrased. Based on your scan settings we also provide insight on how much of the text you are not scanning for plagiarism (Omitted words).

Identical

One to one exact word matches. Learn more

# Minor Changes

Words that hold nearly the same meaning but have a change to their form (e.g. "large" becomes "largely"). <u>Learn more</u>

#### Omitted Words

The portion of text that is not being scanned for plagiarism based on the scan settings. (e.g. the 'Ignore quotations' setting is enabled and the document is 20% quotations making the omitted words percentage 20%) <u>Learn more</u>

#### Copyleaks Internal Database

Our Internal Database is a collection of millions of user-submitted documents that you can utilize as a scan resource and choose whether or not you would like to submit the file you are scanning into the Internal Database. <u>Learn more</u>

#### Filtered and Excluded Results

The report will generate a complete list of results. There is always the option to exclude specific results that are not relevant. Note, by unchecking certain results, the similarity percentage may change. <u>Learn more</u>

#### Current Batch Results

These are the results displayed from the collection, or batch, of files uploaded for a scan at the same time. Learn more

# **\_Q** Plagiarism Detection Results: (1)



(magnetical de la magnetical de la magne

https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf

 $Maida, Esteban\,Gabriel\,;\,Pacienzia, Juli\'an\,Metodolog\'as\,de\,desarrollo\,de\,software\,Tesis\,de\,Licenciatura\,en\,Sistemas\,y\,Computaci\'on\,Facul...$ 

0.2%

REPUBLICA DOMINICANAUNIVERSIDAD AUTONOMA DE SANTO DOMINGOFACULTAD DE CIENCIASEscuela de InformáticaTITULO DEL PROYECTODISEÑO Y DESARROLLO DE UN SISTEMA WEB EN PYTHON PARALA GESTIÓN DE PAGOS E INGRESOS DE CURSOS YDIPLOMADOS EN LA UNIDAD DE EDUCACIÓN CONTINUADA DELA FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES (FCES)SustentantesDeivi Alberto, Turbí Lachapellose Antonio, Ventura MontasYonger Adonis, Suero JeanCoordinadorLic. Eddy BritoAsesor Metodológico Maestro José Manuel Amado Peralta Santo Domingo, República Dominicana, Distrito Nacional febrero del año 2025 IREPUBLICA DOMINICANAUNIVERSIDAD AUTONOMA DE SANTO DOMINGOFACULTAD DE CIENCIASEscuela de InformáticaPRESENTACIÓNTITULO DEL PROYECTODISEÑO Y DESARROLLO DE UN SISTEMA WEB EN PYTHON PARALA GESTIÓN DE PAGOS E INGRESOS DE CURSOS YDIPLOMADOS EN LA UNIDAD DE EDUCACIÓN CONTINUADA DELA FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES (FCES)SustentantesDeivi Alberto, Turbí Lachapel......DE2214Jose Antonio, Ventura Montas......100346945Yonger Adonis, Suero Jean......100363860CoordinadorLic. Eddy BritoAsesor MetodológicoMaestro José Manuel Amado PeraltaSanto Domingo, República Dominicana, Distrito Nacionalfebrero del año 2025 IITEMADISEÑO Y DESARROLLO DE UN SISTEMA WEB ENPYTHON PARA LA GESTIÓN DE PAGOS E INGRESOS DECURSOS Y DIPLOMADOS EN LA UNIDAD DE EDUCACIÓNCONTINUADA DE LA FACULTAD DE CIENCIASECONÓMICAS Y SOCIALES (FCES) iiiFICHA GENERAL DEL PROYECTOInformación principal y autorización del proyectoFecha:09/08/2024Nombre del Proyecto:DISEÑO Y DESARROLLO DE UN SISTEMAWEB EN PYTHON PARA LA GESTIÓN DEPAGOS E INGRESOS DE CURSOS YDIPLOMADOS EN LA UNIDAD DEEDUCACIÓN CONTINUADA DE LAFACULTAD DE CIENCIAS ECONÓMICAS YSOCIALES (FCES)Coordinador del proyecto: Mtro. Eddy BritoAsesores del proyecto: Metodológico: Mtro. José Manuel AmadoContenido: lamel Matos FelizMiembros del proyecto:Deivi Alberto, Turbí LachapelJosé Antonio, Ventura MontasYonger Adonis, Suero JeanFecha de inicio del proyecto:09/10/2024Fecha de finalización:mes/dia/añoNecesidad del proyecto:La creciente demanda de optimización y modernización en el DISEÑO YDESARROLLO DE UN SISTEMA WEB EN PYTHON PARA LA GESTIÓN DEPAGOS E INGRESOS DE CURSOS Y DIPLOMADOS EN LA UNIDAD DEEDUCACIÓN CONTINUADA DE LA FACULTAD DE CIENCIAS ECONÓMICAS YSOCIALES (FCES). Actualmente, la administración manual de estos flujosfinancieros es ineficiente y propensa a errores, dificultando la toma de decisionesy la transparencia. La implementación de un sistema web automatizadocentralizará la información, mejorará la trazabilidad y comunicación, y permitirá ala institución enfrentar los desafíos actuales con mayor competitividad. Tabla 1 Ficha General ivAGRADECIMIENTOSQuiero dar gracias, en primer lugar, a Dios por haberme brindado la fortaleza y laclaridad mental para concluir este proyecto con éxito. Sin su orientación, este caminohabría sido mucho más difícil. Agradezco a la Universidad Autónoma de Santo Domingo (UASD), a la Facultad deCiencias y la Escuela de Informática, por proporcionarme las instalaciones y el apoyoacadémico necesarios para desarrollar esta investigación. Quiero destacar el invaluable respaldo del Lic. Eddy Brito, coordinador del proyecto, quien con su experiencia y generosidad fue una pieza clave para la culminación deeste trabajo. Su orientación constante fue fundamental para superar los desafíos. Un agradecimiento especial al maestro losé Manuel Amado Peralta, cuyoasesoramiento metodológico fue indispensable para darle forma a mis ideas y darestructura a este trabajo. Y a mi asesor de contenido, lamel Matos Feliz, por su apoyoy aporte a la mejora de mi proyecto. A mi familia, por su paciencia, cariño y por brindarme el apoyo emocional necesarioen los momentos de incertidumbre. Su amor me impulsó a seguir adelante. Enespecial, agradezco a mi madre Denia M. Lachapel y a mis hermanos Lic. Jairo Lachapel, Ing. Jorge Lachapel y Lic. Yenny Lachapel por su gran apoyo como familia. Finalmente, agradezco profundamente a mis compañeros de equipo, José AntonioVentura Montas y Yonger Adonis Suero Jean, por su esfuerzo y colaboraciónconstante. Este trabajo es fruto del compromiso colectivo y refleja nuestro esfuerzoconjunto. Deivi A. Turbi Lachapel

vAGRADECIMIENTOSQuiero expresar mi más profundo agradecimiento a Dios, quien me dio laperseverancia, sabiduría y la fortaleza

Autónoma de Santo Domingo (UASD), a la Facultad deCiencias y la Escuela de Informática, por ofrecerme un entorno propicio y los
recursosesenciales para llevar a cabo mi investigación.Un agradecimiento especial al Lic. Eddy Brito, quien, como coordinador del
proyecto,me brindó su constante apoyo y orientación. Su compromiso fue esencial para queeste trabajo llegara a buen término. No puedo
dejar de mencionar al maestro José Manuel Amado Peralta, mi asesormetodológico, cuyo conocimiento y paciencia me ayudaron a
estructurar mis ideas demanera clara y coherente.A mi familia, por estar siempre a mi lado, brindándome su amor, apoyo y motivaciónen
cada paso de este recorrido. Sin su confianza y respaldo, el camino habría sidomucho más difícil. Finalmente, agradezco a mis compañeros
de equipo, Deivi Alberto Turbí Lachapel yYonger Adonis Suero Jean, por su entrega y compromiso en este proyecto. Sucolaboración y
dedicación fueron fundamentales para alcanzar este logro. Jose Antonio Ventura Montas
viAGRADECIMIENTOSEn primer lugar, quiero agradecer a Dios por haberme otorgado la fortaleza, sabiduríay paciencia necesarias para
concluir con éxito este proyecto. Su guía fue clave entodo momento. Agradezco de corazón a la Universidad Autónoma de Santo Domingo
(UASD) y a laFacultad de Ciencias, por brindarme un entorno de aprendizaje adecuado y porproporcionarme las herramientas necesarias
para llevar a cabo esta investigación.Un especial reconocimiento al Lic. Eddy Brito, quien, como coordinador, me brindóuna guía
inestimable a lo largo de todo el proceso. Su disposición para ayudarme enlos momentos críticos fue invaluable.También agradezco
profundamente al maestro José Manuel Amado Peralta, por suincansable apoyo en la metodología de la investigación. Sus consejos me
ayudarona dar forma a mis ideas y a lograr un resultado final claro y organizado. A mi familia, por su incondicional apoyo, por su
comprensión y por ser una fuente deánimo constante. Gracias por estar siempre a mi lado, incluso en los momentos
másdesafiantes. Finalmente, agradezco a mis compañeros de equipo, Deivi Alberto Turbí Lachapel yJose Antonio Ventura Montas, por su
esfuerzo, dedicación y compromiso. Su trabajoconjunto fue esencial para la realización de este proyecto. Yonger Adonis Suero Jean
viiRESUMEN EJECUTIVOEste proyecto tiene como objetivo desarrollar un sistema web que automatice lagestión de pagos e ingresos de
los cursos y diplomados de la Unidad de EducaciónContinuada de la Facultad de Ciencias Económicas y Sociales (FCES) de la
UASD.Actualmente, estos procesos se realizan de manera manual, lo que ocasionaineficiencias y errores. Con la implementación de la
solución propuesta, secentralizarán los procesos administrativos y financieros, optimizando la gestión de lospagos y mejorando la
experiencia de los usuarios.El sistema está diseñado para automatizar tareas como el registro de pagos y lageneración de reportes
financieros detallados. Personal administrativo sebeneficiarán al acceder a una plataforma digital accesible desde cualquier dispositivocon
conexión a internet. Con la digitalización de los procesos, problemas como la faltade transparencia, trazabilidad y los errores
administrativos serán minimizados, logrando una mayor eficiencia en la gestión financiera de la unidad. El desarrollo del sistema se llevará a
cabo utilizando Python y Django para el backend,mientras que MySQL será la base de datos. La interfaz de usuario será implementadacon
Wix. Además, el sistema incluirá módulos esenciales como la autenticación deusuarios, la gestión de pagos, la generación de reportes y
notificaciones automáticas. El proceso de desarrollo abarcará diversas fases, desde el análisis inicial hasta lapuesta en marcha, con un plan
de mantenimiento continuo para asegurar elfuncionamiento a largo plazo. Se espera que el sistema mejore la eficiencia operativa en al
menos un 30%, reduciendo tanto la carga administrativa como los errores humanos. El proyectorequiere una inversión inicial para cubrir el
desarrollo, adquisición de hardware, software, y la infraestructura necesaria, estimando un presupuesto que tambiéncontemple el
mantenimiento a largo plazo. A largo plazo, esta solución representaráuna modernización clave en la gestión financiera de la unidad,
asegurandotransparencia y seguridad en las transacciones.
ÍNDICEPRESENTACIÓN
iTEMAiiFICHA GENERAL DEL PROYECTO
iiiAGRADECIMIENTOSivRESUMEN
EJECUTIVOvii1. Introducción

emocional necesarias para lograr el éxito eneste proyecto. Sin su guía, este logro no habría sido posible. Agradezco a la Universidad

22.1 Descripción Del Proyecto	
	.4
Necesidad del Proyecto	
Proyecto	2.7
Descripción De Los Entregables Del Proyecto73. Equipo De Trabajo.	
	2
Organigrama del proyecto	
114. Descripción De Los Aspectos Técnicos144.1 Requerimientos de hardwa	ire.
	tura
técnica del proyecto	••
194.3.2. Componentes de la Arquitectura	
	.3.5.
Mantenimiento y Escalabilidad	
226. Lista De Actividades	
7. Descripción De Actividades. (eXPRESADOS EN DIAS)	
4710. Matriz De Información. (tiempo en horas)	
	713.
Descripción De Las Limitaciones De Recursos	
	t
6517. Conclusiones Y Recomendaciones	
entrevistas - Preguntas formuladas al cliente)	
	ción
de equipos Máximo)	
79A2.4 (Factura cotización de equipos Recomendado)	
	е
Datos)	
pg. 11. INTRODUCCIÓNEn el presente proyecto se desarrolla un sistema web en Python destinado a la gestiónde pagos e ingresos de lo	s
cursos y diplomados ofrecidos por la Unidad de EducaciónContinuada de la Facultad de Ciencias Económicas y Sociales (FCES) de	
laUniversidad Autónoma de Santo Domingo (UASD).La necesidad de esta herramienta surge debido a la ineficiencia de los	
métodosmanuales utilizados actualmente, que presentan dificultades en la trazabilidad de lastransacciones, la transparencia y la toma d	de
decisiones financieras. El objetivoprincipal es automatizar los procesos relacionados con los pagos y generar informesfinancieros	
detallados, mejorando así la experiencia del personal administrativo. Para ello, se implementará una plataforma accesible desde cualqu	ier
computador conconexión a internet, con altos estándares de seguridad y una interfaz intuitiva quecentralizará toda la información	
financiera y administrativa. El sistema se desarrollará utilizando Python (Django) como tecnología backend, queproporcionará la estruc	tura
y lógica necesarias para gestionar los datos y procesosdel sistema. En el frontend, se utilizará Wix, facilitando una interfaz amigable pa	
losusuarios finales. Además, se emplearán bases de datos MySQL para almacenar yorganizar la información generada por el sistema. El	

monitoreo y la estabilidad delsistema serán asegurados mediante el uso de UptimeRobot, una herramienta demonitoreo que garantizará la disponibilidad del servicio y la pronta respuesta anteposibles incidentes. El documento detalla los aspectos clave del desarrollo del sistema, incluyendo laplanificación, los costos, los riesgos, las herramientas utilizadas y el seguimiento delprogreso, les invitamos a conocer más de este proyecto a continuación.

pg. 22. DESCRIPCIÓN GENERAL DEL PROYECTO.2.1 Descripción Del Proyecto. Nuestra investigación se enfoca en el desarrollo de un sistema web innovador para lagestión eficiente de los procesos financieros en la Unidad de Educación Continuada. Este sistema tiene como objetivo mejorar la eficiencia y la transparencia en la gestiónde pagos e ingresos relacionados con los cursos y diplomados ofrecidos, abarcandodesde el registro de pagos hasta la generación de informes financieros detallados. Como equipo de desarrollo, hemos diseñado un sistema web que será implementadoutilizando tecnologías modernas de desarrollo, garantizando su accesibilidad yfacilidad de uso. Los usuarios administrativos, podrán acceder al sistema desdecualquier computador con conexión a internet, lo que facilitará la administración depagos, consulta del estado financiero y el seguimiento de las transacciones en tiemporeal. Además, el sistema contará con módulos especializados para cubrir áreas clavecomo la gestión de pagos, seguimiento de ingresos, control de cuentas por cobrar yla generación automática de reportes financieros. Entre las características clave del sistema se incluyen:a) Control de Usuarios: Control de acceso según roles, permitiendo que elpersonal administrativo y contable tengan acceso a las funcionalidades queles corresponden.b) Automatización de Procesos: Reducción de tareas manuales mediantela automatización de procesos como la conciliación de pagos y generaciónde comprobante.c) Centralización de Información: Consolidación de toda la informaciónfinanciera en una plataforma única para mejorar la precisión y accesibilidadde los datos.d) Interfaz Amigable: Diseño intuitivo que facilita la navegación y el uso delsistema, tanto para usuarios internos como externos.e) Seguridad y Privacidad: Implementación de altos estándares deseguridad para proteger la información financiera y personal de los usuariosadministrativos.

pg. 3Estas características permitirán a la Unidad Continuada mejorar su eficienciaoperativa, reducir errores en la gestión de pagos e ingresos, y optimizar la experienciatanto de los participantes como del personal administrativo. El sistema no solocontribuirá a una gestión administrativa y financiera más efectiva, sino que tambiénfortalecerá la transparencia y la confianza en los procesos administrativos yfinancieros de la Unidad.2.2 Objetivo General. Diseñar y desarrollar un sistema web en python para la gestión de pagos e ingresosde cursos y diplomados en la unidad de educación continuada de la Facultad DeCiencias Económicas Y Sociales (FCES)2.3 Objetivo Específicos. Nuestro proyecto tiene como objetivo alcanzar los siguientes objetivos específicos: 

Diseñar la arquitectura del sistema web para la gestión de pagos e ingresos. 

Desarrollar un módulo de registro y seguimiento de pagos. 

Implementar un sistema de generación de reportes financieros. 

Integrar notificaciones para participantes y administradores. 

Garantizar la accesibilidad del sistema web para todos los participantes. 

Realizar pruebas de rendimiento y seguridad del sistema. 

A Necesidad del Proyecto. 

La Unidad de Educación Continuada de la Facultad de Ciencias Económicas ySociales (FCES) actualmente carece de un sistema automatizado para la gestión delos pagos e ingresos de los cursos y diplomados. 

Todos los procesos se realizanmanualmente, lo que implica un gran esfuerzo por parte del personal administrativo, incrementa la probabilidad de errores humanos, y dificulta el seguimiento preciso delos ingresos. 

Esta situación provoca retrasos en la generación de reportes financieros, falta de transparencia en los procesos y posibles pérdidas de información crítica.

pg. 4Ante este panorama, se hace indispensable el diseño y desarrollo de un sistema weben Python que automatice y centralice la gestión de pagos e ingresos. Esto permitiráoptimizar los procesos, mejorar la eficiencia operativa, reducir errores y brindar unservicio más eficiente y accesible tanto para los participantes como para el equipoadministrativo. A continuación, destacamos las principales razones que justifican laimplementación de un sistema web para la gestión de pagos e ingresos: ✓ Falta de automatización: Los procesos de gestión de pagos e ingresos serealizan de forma manual, lo que aumenta la carga administrativa. ✓ Propensión a errores humanos: El manejo manual de los datos incrementalas posibilidades de errores en los registros y cálculos. ✓ Ineficiencia operativa: La administración manual ralentiza los procesos, loque ocasiona demoras en la generación de reportes financieros. ✓ Falta de transparencia: No hay un

seguimiento claro y preciso de los pagose ingresos, lo que dificulta la supervisión financiera. ✓ Riesgo de pérdida de información: La falta de un sistema centralizadopuede resultar en pérdidas o inconsistencias en los datos financieros. ✓ Dificultad en la generación de reportes: Crear reportes financieros precisosde manera manual es un proceso tedioso y poco eficiente. 2.5 Antecedentes Del Proyecto. El desarrollo de sistemas web para la gestión administrativa en institucioneseducativas ha sido objeto de interés en los últimos años, especialmente debido alaumento en la oferta de cursos y diplomados, y la creciente demanda por solucionesdigitales que permitan automatizar procesos complejos. Numerosos centros deformación, tanto a nivel nacional como internacional, han implementado plataformasweb que facilitan la gestión de sus programas académicos y administrativos, mejorando la eficiencia operativa y la satisfacción de los usuarios. En este contexto, la Unidad de Educación Continuada ha operado tradicionalmentecon sistemas de gestión manuales o herramientas no integradas, lo que ha generadoproblemas recurrentes, como la duplicación de esfuerzos, la desorganización de lainformación financiera y la dificultad para mantener una comunicación fluida entre losdistintos actores involucrados (estudiantes, personal administrativo y docentes).

pg. 5Existen varios ejemplos de instituciones que han modernizado sus procesosadministrativos mediante sistemas web integrales. Por ejemplo, algunasuniversidades han implementado plataformas que abarcan desde la inscripción ypago de cursos hasta el seguimiento del rendimiento académico y la administraciónde recursos financieros. Estas soluciones no solo han optimizado los procesosoperativos, sino que también han contribuido a mejorar la calidad del servicio alproporcionar una plataforma accesible para el personal administrativo, los estudiantesy los docentes. Con los avances en las tecnologías de la información, hoy es posible desarrollarsistemas personalizados que se adapten a las necesidades específicas de cadainstitución educativa. En este sentido, nuestro proyecto se nutre de estosantecedentes y se adapta a las particularidades de la Unidad de EducaciónContinuada, que requiere una gestión eficaz de los pagos e ingresos de sus cursosy diplomados. La implementación de un sistema web se inspira en los aprendizajesobtenidos de otras instituciones, incorporando innovaciones que mejorarán laaccesibilidad y eficiencia en la administración de los procesos internos. A nivel local, algunas instituciones han comenzado a adoptar plataformas de gestiónen línea, aunque estos esfuerzos aún no se han consolidado en áreas de educacióncontinuada, que suelen tener requisitos más complejos debido a la naturaleza diversade sus programas. Nuestro proyecto tiene como objetivo implementar un sistemaadaptado específicamente a las necesidades de la Unidad de Educación Continuada, aplicando buenas prácticas en el desarrollo de software y aprendiendo tanto de los éxitos como de los desafíos enfrentados por otras instituciones educativas.2.6 Alcance Del Proyecto.El proyecto de desarrollo del Sistema Web para la Gestión de Pagos e Ingresosen la Unidad de Educación Continuada tiene como objetivo principal el diseño eimplementación de una plataforma web integral que abarque todos los aspectos claverelacionados con la gestión de pagos e ingresos de los cursos y diplomados. Estaplataforma estará enfocada en optimizar los procesos administrativos de la unidad, mejorando la experiencia de los usuarios y facilitando una gestión más eficiente. pg. 6Aspectos clave del proyectoDiseño e implementación del sistema web: Se desarrollará una plataformaaccesible a través de un navegador web, que permitirá a los usuarios realizargestiones relacionadas con la inscripción, el pago de cursos, y la administración deingresos de manera automatizada. El sistema estará adaptado a las necesidadesespecíficas de la Unidad de Educación Continuada, garantizando una experienciaoptimizada y simplificada para el personal administrativo.Módulos principales≻ Control de usuarios: Se implementará un sistema de autenticación yautorización que permitirá al personal administrativo el ingreso a la plataformacon roles y permisos diferenciados, asegurando la privacidad y seguridad delos datos financieros. ➤ Inscripción y gestión de pagos: Los administradores con acceso a modificarpodrán inscribir en cursos, gestionar pagos y consultar el estado de lastransacciones. También podrán supervisar y controlar el flujo de ingresos, asícomo realizar conciliaciones de los pagos recibidos. ➤ Notificaciones y comunicación interna: El sistema enviará notificacionesautomáticas cuando se realicen pagos de inscripciones. Esto mejorará lacomunicación y facilitará la gestión de los trámites entre los usuarios. > Generación de reportes financieros: El sistema permitirá generar reportesdetallados sobre los ingresos generados por los cursos y diplomados, facilitando la toma de decisiones y el análisis de indicadores clave para lagestión financiera de la unidad. Seguridad y privacidad de datos El sistema estará diseñado con las más altas medidas de seguridad para

garantizarla confidencialidad y protección de los datos financieros y personales de los usuarios. Esto incluirá autenticación segura, encriptación de datos sensibles y control de accesobasado en roles, lo cual es crucial para el manejo seguro de transacciones.

pg. 7Interfaz amigable y accesible el sistema contará con una interfaz de usuario intuitiva y accesible, diseñada paragarantizar una experiencia de usuario fluida y sencilla para todos los perfilesadministrativos. Implementación y pruebas e realizarán pruebas exhaustivas del sistema para asegurar su correctofuncionamiento en diferentes dispositivos y navegadores web, garantizando suestabilidad y rendimiento. Además, se ofrecerá capacitación al personaladministrativo sobre el uso de la plataforma, asegurando su correcta adopción. Limitaciones geográficas y de acceso Aunque el sistema está diseñado para la Unidad de Educación Continuada, al seruna plataforma en línea, los usuarios podrán acceder desde cualquier lugar conconexión a internet, lo que amplía considerablemente su alcance. Mantenimiento y actualizaciones el proyecto incluirá una fase de mantenimiento continuo, en la cual se corregiránerrores y se realizarán mejoras necesarias para garantizar el buen funcionamiento delsistema a largo plazo. Asimismo, se implementará un plan de actualizaciones regulares que permitirá incorporar nuevas funcionalidades según las necesidades mergentes de la unidad. 2.7 Descripción De Los Entregables Del Proyecto. El proyecto de desarrollo del Sistema Web para la Gestión de Pagos e Ingresos de la Unidad de Educación Continuada contempla la entrega de los siguientes productos y servicios que garantizarán la implementación exitosa y el buenfuncionamiento de la plataforma:

pg. 8Entregables ➤ E01 Documentación del proyecto (backend): Se entregará un informecompleto que detallará todos los aspectos del sistema, incluyendo ladescripción funcional y no funcional, la arquitectura del sistema, los módulos yfuncionalidades desarrolladas, así como el plan de implementación y laspruebas realizadas. (Link)1 ➤ E02 Diseño de la interfaz de usuario (HTML): Se entregará un diseñodetallado de la interfaz de usuario que incluirá la estructura de las páginas, lanavegación, la disposición de la información y la interacción con los usuariosadministrativo. (Link)2 ➤ E03 Código fuente del sistema: Se entregará el código fuente completo,incluyendo todos los archivos necesarios para la implementación y elfuncionamiento del sistema. Este código será accesible para futurasactualizaciones o modificaciones. (Link)2 ➤ E04 Implementación del sistema: El sistema se implementará en un servidorweb, configurado para estar disponible para los usuarios de la Unidad deEducación Continuada, garantizando su acceso desde cualquier computadorcon conexión a internet. (Link)3 ➤ E05 Pruebas y validación: Se realizarán pruebas exhaustivas para verificarel correcto funcionamiento del sistema y asegurar que cumpla con losrequisitos planteados. Estas pruebas incluirán validaciones funcionales, deseguridad y de rendimiento. (Link)2 ➤ E06 Manual de usuario: Se entregará un manual detallado que incluiráinstrucciones paso a paso sobre cómo utilizar el sistema, con capturas depantalla y explicaciones claras de cada funcionalidad. (Link)21 https://finalapi.up.railway.app/api/docs/2 https://github.com/deivilachapel/entregables3 https://wilberthvers03-wixsite-com.filesusr.com/html/4eb7fl\_bbef188c7ac615ed25dadac44aaa81e5.html

pg. 9Cronograma de entregables (expresado en semanas)Nombres Entregables CantidadDocumentación del proyecto (backend) 1Diseño de la interfaz de usuario (HTML) 1Código fuente del sistema 2Implementación del sistema 3Pruebas y validación 3Manual de usuario 4Responsables de los entregables. El equipo de desarrollo será responsable de la entrega de la documentacióndel proyecto, el diseño de la interfaz de usuario, el código fuente del sistema, la implementación, las pruebas y validación, así como del plan demantenimiento y actualizaciones. El personal administrativo será responsable de participar en la capacitacióny de utilizar el sistema de acuerdo con las indicaciones proporcionadas. Tanto el equipo de desarrollo como el personal administrativo colaboraránen la creación del manual de usuario, asegurando que sea claro y accesiblepara todos los usuarios.

pg. 10No entregables N.o Descripción de la actividad Breve descripción Motivo 1 Entrenamientos A Los Participantes Del proyecto Se realizan reuniones y seexplican tanto la metodología como las tecnologías que seutilizaran en el proyecto. Se haría en una segunda entrega por falta de tiempo 2 Carga de información Por QR Automático Se lee la información de lacedula de identidad delusuario desde el QRSe haría en la segunda entrega por falta de tiempo 3 Análisis Y Monitoreo Del Software Se verifica el debido funcionamiento del sistema y su mantenimiento necesario. Se haría en la segunda entrega por falta de tiempo Tabla 2 No entregables 3. EQUIPO DE TRABAJO. 3.1 Miembros

Alberto Turbí Lachapel ✓ Asesor: José Amado ✓ Encargado de Base de Datos: Deivi Alberto Turbí Lachapel ✓ Desarrollador Backend (Django): Deivi Alberto Turbí Lachapel y JoséAntonio Ventura Montás√ Desarrollador Frontend (Wix): José Antonio Ventura Montás√ Diseñador UI/UX: José Antonio Ventura Montás ✓ Tester/QA: Yonger Adonis Suero Jean pg. 113.2 Organigrama del proyecto.3.3 Descripción De Las Funciones Y Responsabilidades. Responsable Puesto Descripción Lic. Eddy Brito Gerente deProyectosResponsable de la planificación, supervisión y control delproyecto, asegurando que el equipo cumpla con los plazosy los objetivos establecidos. Coordina la comunicaciónentre los miembros del equipo y gestiona los recursosnecesarios para completar el proyecto de manera efectiva. Maestro José Manuel Amado Peralta Asesor de Metodológico Brinda apoyo y orientación estratégica al equipo dedesarrollo, asegurando que las decisiones tomadas esténalineadas con los objetivos del proyecto y las mejoresprácticas de la industria. pg. 12Responsable Puesto Descripciónlamel MatosFelizAsesor deContenidoColabora estrechamente con el equipo de desarrollo, brindando orientación sobre la correcta formulación de losdocumentos, la adecuación del lenguaje, y la integraciónde fuentes y referencias relevantes. Además, proporcionaretroalimentación crítica para mejorar la claridad yprecisión de los informes y garantizar que estos reflejenadecuadamente los avances y resultados obtenidos. Deivi Alberto Turbí Lachapel Arquitecto de Software Diseña la estructura técnica de un sistema, tomandodecisiones sobre las tecnologías, herramientas ymetodologías que se utilizarán en el desarrollo. Es elresponsable de asegurar que el sistema sea escalable, eficiente y fácil de mantener. Deivi Alberto Turbí Lachapel Encargado de Base deDatosResponsable del diseño, implementación y mantenimientode la base de datos. Garantiza la integridad, seguridad yeficiencia en el almacenamiento y acceso a la información, gestionando los backups y optimizando las consultas. Deivi Alberto Turbí Lachapel José Antonio Ventura Montás Desarrollador Backend (Diango) Encargado del desarrollo del servidor, la lógica de negocio y la creación de APIs para la interacción con la base dedatos. Se asegura de que el sistema backend seaeficiente, escalable y seguro. José AntonioVenturaMontásDesarrollador Frontend(Wix)Responsable de la implementación de la interfaz deusuario en el frontend, utilizando Wix. Asegura la correctavisualización y usabilidad de la plataforma en todos losdispositivos, optimizando el rendimiento y la interaccióndel usuario. José Antonio Ventura Montás Diseñador UI/UXD iseña la experiencia de usuario y la interfaz gráfica, garantizando que sea intuitiva, atractiva y fácil de usar. Seenfoca en crear una navegación fluida y una estética visualcoherente con la marca y el objetivo del proyecto.

del proyecto.El equipo encargado de la realización del proyecto está conformado por los siguientesmiembros: ✓ Project Manager: Deivi

pg. 13Responsable Puesto DescripciónDeivi AlbertoTurbí LachapelAnalistaFuncionalActúa como un intermediario entre los usuarios finales y elequipo técnico, asegurando que las funcionalidades delsistema cumplan con los objetivos del negocio. José

AntonioVenturaMontásAnalista deSistemasEs responsable de investigar, evaluar y diseñar solucionestecnológicas que optimicen el rendimiento de los sistemasinformáticos. Su función principal es analizar los requisitosdel cliente, traducirlos en especificaciones técnicas ycolaborar en el diseño de soluciones eficientes. Yonger AdonisSuero JeanAnalista deProcesosSe encarga de estudiar y mejorar los procesos de negociode una organización. Su objetivo es identificar ineficienciasy proponer mejoras para optimizar el flujo de trabajo yaumentar la productividad. Yonger AdonisSuero Jean Tester/QAEncargado de realizar pruebas de calidad en el sistemapara identificar errores, asegurando que todas lasfuncionalidades del sistema funcionen correctamente. Seenfoca en la detección de fallos y el aseguramiento de lacalidad antes de la implementación final. Proporcionasoporte técnico a los usuarios finales del. Tabla 3 Funciones Y Responsabilidades

pg. 144. DESCRIPCIÓN DE LOS ASPECTOS TÉCNICOS.4.1 Requerimientos de hardware. Para garantizar el correcto funcionamiento del Sistema Web para la Gestión dePagos e Ingresos de la Unidad de Educación Continuada, es necesario disponerde una infraestructura de hardware adecuada tanto para el desarrollo como para laimplementación y uso del sistema. A continuación, se describen los requerimientos de hardware necesarios en cada fase del proyecto: Servidor de desarrollo y pruebas: ✓ Procesador: Intel Core i5 o equivalente, con al menos 4 núcleos y 3.0 GHz develocidad de reloj. ✓ Memoria RAM: 8 GB o superior, para permitir la ejecución de múltiples procesos simultáneamente durante el desarrollo y pruebas. ✓ Almacenamiento: 256 GB de disco duro SSD o superior, para

asegurar lavelocidad en la carga y procesamiento de archivos del proyecto. Conectividad a Internet: Conexión de banda ancha estable con al menos 10Mbps de velocidad de descarga y carga. Sistema operativo: Windows 10 o superior / Ubuntu 20.04 LTS o superior. pg. 15Servidor de producción: Procesador: Intel Xeon E5 o superior, o AMD EPYC, con al menos 8 núcleospara manejar las solicitudes de varios usuarios de manera eficiente. Memoria RAM: 16 GB o superior, para garantizar que el sistema puedagestionar múltiples usuarios de manera fluida, así como transacciones yconsultas a la base de datos. Almacenamiento: 512 GB de disco duro SSD o superior, asegurandosuficiente espacio para almacenar los datos de usuarios, registros de pagos yotros documentos administrativos. Conectividad a Internet: Conexión dedicada con al menos 50 Mbps develocidad de carga y descarga, para soportar las operaciones en línea devarios usuarios simultáneamente. Sistema operativo: Linux Ubuntu Server 20.04 LTS o superior / CentOS 8, optimizados para servidores web y base de datos. Soporte para backups automáticos: Un sistema de almacenamientosecundario de al menos 1 TB para realizar copias de seguridad periódicas. Equipos de los usuarios finales (personal administrativo): Procesador: Intel Core i3 o equivalente, con al menos 2 núcleos. Memoria RAM: 4 GB o superior. Almacenamiento: 128 GB HDD o SSD. Conectividad a Internet: Conexión estable con al menos 5 Mbps de velocidadde descarga para acceder a la plataforma sin interrupciones. Navegador web: Últimas versiones de Google Chrome, Mozilla Firefox, o Microsoft Edge.

pg. 16Otros requerimientos de hardware: ➤ Dispositivos de respaldo y seguridad: Para proteger la infraestructura delservidor, se recomienda contar con un sistema de alimentaciónininterrumpida (UPS) y un servidor de respaldo para casos de fallo crítico odesastres naturales. ➤ Sistema de protección contra incendios y control climático: En el centrode datos, se requiere contar con un ambiente controlado, tanto en términos detemperatura como de seguridad, para evitar sobrecalentamiento y otros dañosa los equipos.4.2 Requerimientos de software. Para el desarrollo e implementación del Sistema Web para la Gestión de Pagos elngresos de la Unidad de Educación Continuada, que utiliza Django comobackend, Wix como frontend, y MySQL para la base de datos, se deben considerarlos siguientes requerimientos de software: Software para el desarrollo ➤ Sistema operativo: Windows 10 o superior, o Ubuntu 20.04 LTS o superior, dependiendo del entorno dedesarrollo preferido. ➤ Entorno de desarrollo integrado (IDE): Visual Studio Code, para facilitar la programación y gestión del código en Python(Django).

pg. 17 > Frameworks y herramientas: ✓ Django 4.x: Framework de Python para el desarrollo del backend, gestiónde la lógica del negocio y la interacción con la base de datos. ✓ Wix: Plataforma de desarrollo frontend utilizada para la creación de lainterfaz de usuario. Aunque Wix es una plataforma no tradicional paradesarrollo frontend, se debe integrar con el backend de Django a través deAPIs o herramientas disponibles. > Base de datos: MySQL 8.x: Sistema de gestión de bases de datos relacional, desplegado paraalmacenar y gestionar los datos del sistema, incluyendo usuarios, pagos y registros. > Lenguajes de programación: ✓ Python 3.9 o superior para el desarrollo del backend con Django. ✓ JavaScript (ES6+) para el frontend, principalmente en la integración conWix si es necesario. ✓ HTML5 y CSS3: Para diseño y estilos de la interfaz de usuario, especialmente si Wix permite modificaciones de código. > Control de versiones: Git: Herramienta para la gestión del código fuente y colaboración en el proyecto. Plataformas como GitHub o GitLab se utilizarán para el almacenamiento del código. > Servidor web local: XAMPP: Aunque XAMPP es generalmente utilizado para entornos de desarrollo PHP, se puede utilizar para pruebas locales de MySQL y conexiones de backend. Software para la implementación en producción > Sistema operativo del servidor: Ubuntu Server 20.04 LTS o CentOS 8: Recomendado para la implementación delbackend Django y el servicio de base de datos MySOL.

pg. 18 > Servidor web:Apache 2.4 o Nginx: Servidores web compatibles con Django y la integración con labase de datos MySQL. > Base de datos en producción:MySQL 8.x: Configurado para manejo eficiente de datos en un entorno de producción. > Lenguajes y frameworks de servidor: ✓ Python 3.9+ con Django 4.x para el backend. ✓ WIX para cualquier integración frontend. > Sistema de gestión de backups:rsync y cron jobs: Para realizar copias de seguridad periódicas de la base de datosy configuraciones del servidor. > Certificados de seguridad:SSL/TLS: Para cifrar la comunicación entre los usuarios y el servidor, protegiendo losdatos sensibles. Software de seguridad y monitoreo > Firewall y seguridad:UFW (Uncomplicated Firewall) en Ubuntu o Firewalld en CentOS para proteger elservidor contra accesos

no autorizados. > Monitoreo del sistema: Uptimerobot: Para la monitorización del rendimiento y la estabilidad del sistema. Software para los usuarios finales (personal administrativo) > Navegador web compatible: Últimas versiones de Google Chrome, Mozilla Firefox, o Microsoft Edge para unaexperiencia óptima al acceder al sistema web.

pg. 19 >> Requisitos adicionales: Correo electrónico para la recepción de notificaciones y comunicados delsistema. PDF Reader como Adobe Acrobat Reader para visualizar documentosgenerados por el sistema. 4.3 Arquitectura técnica del proyecto. La arquitectura técnica del Sistema Web para la Gestión de Pagos e Ingresos dela Unidad de Educación Continuada se diseña para integrar eficientemente elbackend desarrollado con Python (Django), el frontend creado en Wix, y la base dedatos MySQL. La arquitectura sigue un enfoque de arquitectura en capas yarquitectura de microservicios, proporcionando una solución robusta, escalable ysegura. A continuación, se describen los componentes y su interconexión: 4.3.1. Arquitectura General Frontend (Wix): Wix se utiliza para el diseño de la interfaz de usuario del sistema. Ofrece unaplataforma para crear y gestionar la experiencia visual y la interacción del usuario. Wix se integra con el backend a través de APIs para realizar llamadas a servicios yobtener datos en tiempo real. Backend (Django): Django se encarga de la lógica del negocio, el procesamiento de datos y la gestión de usuarios. Django proporciona APIs RESTful que permiten la comunicación entreel frontend y el backend, facilitando la manipulación de datos y la ejecución deoperaciones administrativas. Base de Datos (MySQL): MySQL es el sistema de gestión de bases de datos relacional que almacena toda lainformación del sistema, incluyendo datos de usuarios, transacciones, y registrosacadémicos. La base de datos está conectada al backend Django medianteconectores y ORM (Object-Relational Mapping).

pg. 204.3.2. Componentes de la ArquitecturaFrontend (Wix): Diseño Visual: Wix proporciona las herramientas para construir la interfaz deusuario, incluyendo la disposición de elementos, estilos y funcionalidadesbásicas. Integración API: Wix utiliza JavaScript para hacer solicitudes HTTP a lasAPIs RESTful proporcionadas por Django, permitiendo la interacción con elbackend. Backend (Django): API RESTful: Django expone endpoints de REST API que permiten al frontendconsultar y actualizar datos en la base de datos. Estos endpoints estánprotegidos con autenticación y autorización. Lógica del Negocio: Django maneja la lógica de negocio, incluyendo lavalidación de datos, el procesamiento de pagos y la gestión de usuarios. Autenticación: Implementación de un sistema de autenticación yautorización para controlar el acceso a las diferentes funcionalidades delsistema. Seguridad: Utiliza middleware y prácticas de seguridad para proteger laaplicación contra vulnerabilidades comunes. Base de Datos (MySQL): Modelo de Datos: Django utiliza ORM para interactuar con MySQL, facilitandola creación, lectura, actualización y eliminación de registros. Estructura de Datos: La base de datos está estructurada en tablas quereflejan los diferentes aspectos del sistema, como usuarios, pagos, cursos ydiplomados. Seguridad de Datos: Implementación de medidas de seguridad para protegerlos datos sensibles, incluyendo cifrado y control de acceso.

pg. 214.3.3. Flujo de Datos1. Solicitud del Usuario: Un usuario realiza una acción en la interfaz de Wix,como registrar un curso, registrar una inscripción o un pago académico. 2. Solicitud al Backend: Wix envía una solicitud HTTP a uno de los endpointsRESTful expuestos por Django. 3. Procesamiento en Django: Django recibe la solicitud, realiza las operacionesnecesarias en la base de datos MySQL y procesa la lógica del negocio. 4. Respuesta al Frontend: Django devuelve una respuesta al frontend con losdatos solicitados o el resultado de la operación. 5. Actualización de la Interfaz: Wix actualiza la interfaz de usuario con lainformación proporcionada por Django. 4.3. 4.

Seguridad y Protección Cifrado de Datos: Utilización de SSL/TLS para cifrar la comunicación entre elfrontend y el backend. ✓

Autenticación y Autorización: Implementación de mecanismos deautenticación y autorización en Django para proteger los endpoints de las APIs. ✓ Protección contra Amenazas: Implementación de medidas de seguridad enel backend para proteger contra ataques comunes como inyección SQL ycross-site scripting (XSS).

pg. 224.3.5. Mantenimiento y Escalabilidad A Monitoreo: Se utilizará UptimeRobot para supervisar el tiempo de actividad yel rendimiento del sistema. UptimeRobot permite configurar alertas paranotificar a los administradores sobre cualquier interrupción o problema en elsistema, facilitando una respuesta rápida para resolver problemas y asegurarla disponibilidad Continuada del servicio. A Mantenimiento: Implementación de un plan de mantenimiento regular paraaplicar actualizaciones y correcciones. Escalabilidad: Diseño modular y

basado en servicios para permitir laexpansión y el manejo de un mayor número de usuarios y transacciones sincomprometer el rendimiento.5. PRESUPUESTO.El presupuesto para el Sistema Web de Gestión de Pagos e Ingresos de la Unidadde Educación Continuada abarca los costos asociados con el desarrollo, implementación y mantenimiento del sistema. A continuación, detallamos elpresupuesto estimado para cada categoría clave:5.1. Costos de Desarrollo (Valores expresados en RD\$)5.1.1. Recursos Humanos:≻ Desarrolladores Backend (Django): ✓ Costo estimado: 93,350 por desarrollador. ✓ Total (por 2 desarrolladores durante 4 semanas): 186,700 > Desarrolladores Frontend (Wix): ✓ Costo estimado: 101,700 por desarrollador. ✓ Total (por desarrollador durante 4 semanas): 101,700 > Diseñador UI/UX: ✓ Costo estimado: 90,000 ✓ Total: 90,000 pg. 23 ≻ Tester/QA: ✓ Costo estimado: 115,000 por tester. ✓ Total (por 1 tester durante 4 semanas): 115,0005.1.2. Herramientas y Licencias: ≻ Licencia de Software:Costos de licencias para herramientas de desarrollo (si es necesario): 12,000.≻ Plataforma Wix:Plan de Wix y posibles aplicaciones adicionales: 5,6005.2. Costos de Implementación5.2.1. Infraestructura y Servidores:≻ Servidor Web:Costo estimado del servidor (AWS o DigitalOcean): 232,000 al año. ≻ Base de Datos:Costo de almacenamiento y gestión de base de datos en MySQL: 150,000 al año.5.2.2. Dominio y Certificado SSL: ➤ Nombre de Dominio:Costo anual: 5,600. ➤ Certificado SSL:Costo anual para cifrado seguro: 4,5005.3. Costos de Mantenimiento5.3.1. Soporte Técnico y Mantenimiento: > Mantenimiento del Sistema: Costo estimado anual para correcciones y actualizaciones: 800,000. ➤ Monitoreo (UptimeRobot):Plan de monitoreo y alertas: 6,000 al año. pg. 245.3.2. Capacitación y Documentación: ➤ Capacitación del Personal Administrativo:Costo de sesiones de capacitación: 70,000. ➤ Documentación y Manuales:Costo de preparación y distribución de manuales: 85,000.5.4. Costos Adicionales5.4.1. Contingencia: ➤ Fondo para Imprevistos:Porcentaje del presupuesto total para imprevistos: 10% del total estimado. 197,1105.4.2. Otros Costos: ➤ Gastos Generales:Costos asociados con reuniones y comunicación: 25,000.Resumen del Presupuesto ✓ Costos de Desarrollo: 400,000 ✓ Costos de Implementación: 392,100 ✓ Costos de Mantenimiento: 961,000 ✓ Costos Adicionales: 25,000Total, Estimado del Presupuesto: 1,753,1256. LISTA DE ACTIVIDADES.6.1 Levantamiento de información.N.o Descripción de la actividad Breve Descripción1 Levantamiento DeRequerimientosObtención de información necesaria paraentender lo que el cliente y los usuarios finalesnecesitan del sistema. Se recopila lainformación de la problemática Tabla 4 Levantamiento de información pg. 256.2 Análisis del sistema. N.o Descripción de la actividad Breve Descripción2 Estudios de FactibilidadSe realiza una evaluación preliminar quedetermina la viabilidad del proyecto desde tresenfoques: técnico (infraestructura y habilidades),económico (costos y beneficios), y operacional (impacto en las operaciones y usuarios). Concluye con un informe que resume resultadosy ofrece recomendaciones para decidir sobre elproyecto. 3 Análisis del Sistema Propuesto Se evalúa la viabilidad y estructura del nuevosistema, describiendo sus objetivos generales yespecíficos, justificando su implementación ydefiniendo la metodología a seguir. Tambiénincluye diagramas de flujo de datos, narrativas delos procesos, y un diccionario de datos. Concluyecon un informe detallado que valida la propuestay su adecuación a los requerimientos delproyecto. 4 Requerimientos Funcionales Son requisitos obligatorios para elfuncionamiento del sistema para satisfacer losrequisitos del cliente. 5 Casos De Usos Documentar cómo los diferentes actoresinteractuarán con el sistema para cumplir con losobjetivos planteados. Tabla 5 Análisis del Sistema pg. 266.3 Diseño FuncionalN.o Descripción de laactividad Breve Descripción6 Diseño FuncionalCrear el diseño que especifica cómo debefuncionar el sistema a nivel de interacción, procesos y componentes, es decir se diseñan lasdiferentes funciones del proyecto. 7 Casos De PruebaDefinir los escenarios de prueba basados en loscasos de uso y los requisitos funcionales, paravalidar que el sistema cumple con los requerimientos, es decir se prueban los casosprevios para determinar si cumplen con lascondiciones aceptables. Tabla 6 Diseño **Funcional** 

pg. 276.4 Diseño del Sistema N.o Descripción de la actividad Breve Descripción 8 Arquitectura Preliminar Diseño de la estructura general del sistema, incluyendo componentes clave como servidores, bases de datos y lógicade negocio, es decir se establece la arquitectura con la quese piensa realizar el proyecto. 9 Diseño Técnico Definición técnica detallada del sistema, incluyendo la selección de tecnologías, plataformas y componentes técnicos específicos, es decir dependiendo del diseño funcional, este se encarga de darle los rasgos técnicos a

este.10 Creación De La BaseDe Datos (SQL)Diseño e implementación de las tablas, relaciones yestructuras necesarias para la base de datos del sistema, esdecir se crea la base de datos del sistema.11Diagramas UML QueDocumentan ElProyectoRepresentación visual de la estructura y comportamiento delsistema utilizando diagramas UML (como diagramas declases, de secuencia, etc.). Son diagramas intuitivos queexplican el funcionamiento del sistema de forma no tantécnica. Tabla 7 Diseño Del Sistema pg. 286.5 Desarrollo del SistemaN.o Descripción de laactividad Breve Descripción12 Cargar Proyecto a GitHupConsiste en subir el código del sistema web a unrepositorio en la plataforma GitHub. Esto implicainicializar un repositorio en local, conectar el proyectocon el repositorio remoto, realizar un commit de losarchivos relevantes y finalmente ejecutar el comandogit push para enviar los cambios al servidor deGitHub. Esta actividad asegura que el proyecto estérespaldado y accesible para el equipo de desarrolloo colaboradores.13Convertir Código a Pythonen Model.pyAdaptar o implementar el modelo de datos en elarchivo model.py, utilizando el lenguaje deprogramación Python. Esta tarea implica definir lasclases y atributos que representan las estructuras dedatos del sistema, asegurando que sean compatiblescon la lógica y la base de datos del proyecto.14Creación del archivoserializar.pyConsiste en diseñar y desarrollar un archivo dentrodel proyecto web que define los serializadores. Estosse utilizan para convertir los datos complejos, comoobjetos de modelos, en formatos simples comoJSON o XML, y viceversa.15Creación del archivoviews.pyEsta actividad consiste en desarrollar el archivodonde se definen las vistas del sistema web. En unframework como Django, el archivo 'views.py'contiene las funciones o clases responsables deprocesar las solicitudes del usuario y devolver las respuestas correspondientes. pg. 29N.o Descripción de laactividad Breve Descripción16Creación del archivo url.py Consiste en desarrollar el archivo encargado dedefinir las rutas o endpoints del sistema web. En unframework como Django, este archivo mapea lasURLs que ingresan los usuarios en el navegador conlas vistas correspondientes en el archivo views.py.17Despliegue del proyecto aGitHupConsiste en subir la versión final o actualizada delproyecto al repositorio remoto en GitHub. Estaactividad implica verificar que todos los cambiosestén confirmados localmente con git commit.18Re-subir proyecto a GitHup/GitHup and commitrealizar el envío de los cambios al repositorioremoto con git push, y asegurarse de que laestructura del proyecto esté bien organizada.19Despliegue de la base dedatos en clever cloudConsiste en configurar y publicar la base de datos delsistema web en la plataforma de alojamiento CleverCloud. Esto incluye la creación de una nuevainstancia de base de datos (por ejemplo, MySQL, PostgreSQL), la importación de los datos oesquemas necesarios, y la configuración decredenciales de acceso.20Despliegue del backend(aplicación de Django):RailwayEsta actividad consiste en configurar y publicar laaplicación web desarrollada con Django en laplataforma Railway. pg. 30N.o Descripción de laactividad Breve Descripción21Creación de las páginas enwixImplica diseñar y desarrollar el sitio web utilizando lasherramientas visuales de la plataforma. Primero, seplanifica la estructura del sitio, definiendo quépáginas serán necesarias, como inicio, productos ocontacto. Luego, se personaliza el diseño mediantearrastrar y soltar, ajustando elementos visuales comoimágenes, colores y texto. Además, se integran datosdinámicos conectando la API del backend, permitiendo mostrar contenido actualizado en tiemporeal.22 Desarrollo De ControladoresY ProgramaciónImplementación del código que gobierna elcomportamiento del sistema, integrando lasfuncionalidades con los módulos técnicos, es decir secodifica el sistema en el lenguaje de programaciónelegido.23 Desarrollo De La Interfaz DeUsuarioCreación y diseño de la parte visual del sistema conla que interactuarán los usuarios, es decir se crea lainterfaz del sistema (menú, módulos, gráficos). Tabla 8 Desarrollo del Sistema pg. 316.6 Pruebas y ValidaciónN.o Descripción de la actividad Breve Descripción24 Ambiente De PruebasPreparación del entorno de pruebas donde seejecutarán los test, es decir se inician las pruebasdel sistema. 25 Pruebas Unitarias Validar cada componente del sistema de maneraaislada para asegurar que funcionancorrectamente, es decir se prueban que todos loscomponentes cumplen con su función de maneracorrecta. 26 Pruebas De Errores DelSoftware Ejecutar pruebas que identifiquen posibles falloso errores en el software, es decir se determina losbug o errores que pueda tener el sistema enfuncionamiento.27 Pruebas De Conexión ConLa Base De DatosValidar que la

conexión entre el sistema y la basede datos funcione sin problemas, es decir severifica la conexión y el debido funcionamiento.28Pruebas

De Aceptación DeUsuarioConfirmar que el sistema cumple con lasexpectativas y necesidades de los usuariosfinales mediante pruebas

reales, es decir losusuarios usan la aplicación y dan sus opinionesacerca de esta. Tabla 9 Prueba y Validación pg. 326.7 Planificación y DocumentaciónN.o Descripción de la actividad Breve Descripción29 PlanificaciónDefinir el cronograma y los plazos para cada fasedel proyecto, garantizando una ejecuciónordenada, es decir se le da forma a lainformación recopilada para elaborar el plan yestablecer el horizonte del proyecto.30 Finalizar Documentación DelSoftware PreliminarCompletar la documentación inicial del sistema, cubriendo aspectos funcionales y técnicos, esdecir es un documento preliminar que sirve deguía para comenzar con la parte técnica delproyecto.31 Reportes De PruebasDocumentar los resultados de las pruebasrealizadas y los incidentes encontrados duranteel proceso de validación, es decir son losposibles inconvenientes que pueda tener elsistema en producción. Tabla 10 Planificación y Documentación 6.8 Entrega y Cierre N.o Descripción de la actividad Breve Descripción 32 Ambiente De Entrega Preparar el entorno final donde se desplegará elsistema para su uso productivo, es decir es ellnicio del proceso de implementación del sistemaen producción.33 Entrega De Documentación YProductoHacer la entrega formal del sistema terminado ysu documentación al cliente, es decir seimplementa el sistema con la debidadocumentación y capacitación del personal.Tabla 11 Entrega Y Cierre pg. 337. DESCRIPCIÓN DE ACTIVIDADES. (EXPRESADOS ENDIAS)1. Levantamiento de Requerimientos ✓ Descripción: Recolectar información para identificar las necesidades delcliente y los usuarios finales. ✓ Duración: 7. ✓ Responsable: Analista de sistemas. ✓ Entregable: Documento de requerimientos.2. Estudios de Factibilidad2.1. Factibilidad Técnica / Descripción: Evaluar si la organización cuenta con la infraestructuratecnológica y habilidades necesarias para llevar a cabo el proyecto. ✓ Duración: 3. ✓ Responsable: Analista de sistemas. 

Entregable: Informe de factibilidad técnica. 

Análisis del Sistema Propuesto 

1. Descripción General 

Descripción: Descripción del sistema propuesto y cómo resolverá lasnecesidades detectadas en los requerimientos. ✓ Duración: 2. ✓ Responsable: Analista de sistemas. ✓ Entregable: Documento de análisis del sistema propuesto. pg. 344. Requerimientos Funcionales 🗸 Descripción: Definir y documentar las funcionalidades que el sistema debecumplir. 🗸 Duración: 5. 🗸 Responsable: Analista funcional. ✓ Entregable: Documento de especificación funcional. 5. Casos de Usos ✓ Descripción: Detallar cómo los usuarios interactúan con el sistema mediantecasos de uso. ✓ Duración: 4. ✓ Responsable: Analista de sistemas. ✓ Entregable: Documento de casos de uso.6. Diseño Funcional / Descripción: Desarrollar el diseño funcional, incluyendo pantallas, flujo dedatos y casos de uso. / Duración: 9. ✓ Responsable: Diseñador UI/UX. ✓ Entregable: Documento de diseño funcional.7. Casos de Prueba ✓ Descripción: Definir casos de prueba para validar que el sistema cumple conlos requisitos. ✓ Duración: 4. ✓ Responsable: Tester/QA. pg. 35Entregable: Matriz de casos de prueba.8. Diseño del sistema / Descripción: Planificar y estructurar la arquitectura del sistema web, incluyendo la definición de los componentes principales, su interacción ylas tecnologías a utilizar. Esto abarca el diseño de la base de datos, losdiagramas de clases, los flujos de información y la interfaz de usuario. Elobjetivo es garantizar un sistema funcional, escalable y fácil de mantener. ✓ Duración: 7. ✓ Responsable: Analista de sistemas y Diseñador UI/UX. ✓ Entregable: Documentación del diseño del sistema, incluyendo diagramasy prototipos de la interfaz.9. Arquitectura Preliminar√ Descripción: Crear una primera versión de la arquitectura del sistema,considerando componentes clave y su interacción. ✓ Duración: 4. ✓ Responsable: Arquitecto de software. ✓ Entregable: Diagrama de arquitectura preliminar.10. Diseño Técnico ✓ Descripción: Definir la estructura técnica del sistema, seleccionandotecnologías e infraestructura. ✓ Duración: 5. ✓ Responsable: Arquitecto de software. ✓ Entregable: Documento de diseño técnico. pq. 3611. Creación de la Base de Datos (SQL.) ✓ Descripción: Diseñar e implementar la base de datos con todas las tablas yrelaciones necesarias. ✓ Duración: 5. ✓ Responsable: Administrador de base de datos. ✓ Entregable: Base de datos operativa.12. Diagramas UML que Documentan el Proyecto ✓ Descripción: Crear diagramas UML para representar gráficamente laestructura del sistema. ✓ Duración: 5. ✓ Responsable: Arquitecto de software. ✓ Entregable: Conjunto de diagramas UML.13. Despliegue del proyecto a GitHub ✓ Descripción:

Subir el proyecto completo al repositorio remoto en GitHub paragarantizar su respaldo y facilitar la colaboración en equipo. Incluye

laconfiguración inicial, realizar commits y ejecutar el push al repositorio. ✓ Duración: 1. ✓ Responsable: Desarrollador. ✓ Entregable:

Proyecto alojado y actualizado en un repositorio de GitHub.

pg. 3714. Convertir código a Python en models.py Descripción: Crear las clases y estructuras del modelo de datos en

Python,representando las entidades del sistema en el archivo models.py. Estoincluye definir atributos, relaciones y métodos según los
requerimientos delsistema web. Duración: 2. Responsable: Desarrollador Backend. Entregable: Archivo models.py con las clases y
atributos correctamentedefinidos.15. Creación del archivo serializers.py Descripción: Diseñar y desarrollar los serializadores que
convierten datoscomplejos en formatos simples (como JSON) y viceversa, permitiendo lacomunicación entre el backend y las aplicaciones
cliente. Duración: 1. Responsable: Desarrollador Backend. Entregable: Archivo serializers.py con los serializadores definidos
yfuncionales.16. Creación del archivo views.py Descripción: Desarrollar las vistas responsables de procesar las solicitudesy devolver las
respuestas correspondientes. Incluye implementar la lógicanecesaria para interactuar con los modelos y serializadores, y gestionar
lasfuncionalidades del sistema. Duración: 2. Responsable: Desarrollador Backend. Entregable: Archivo views.py con las vistas
definidas y funcionales.

pg. 3817. Creación del archivo urls.py Descripción: Definir las rutas del sistema que mapean las solicitudes alas vistas correspondientes.

Esto permite estructurar y organizar el accesoa las diferentes funcionalidades del sistema web. Duración: 1. Responsable: Desarrollador Backend. Entregable: Archivo urls.py con las rutas configuradas y funcionales.18. Re-subir proyecto a GitHub / GitHub and Commit Descripción: Actualizar el repositorio en GitHub con los cambios recientesdel proyecto. Incluye realizar nuevos commits con una descripción adecuadade los cambios y ejecutar el comando push para sincronizar los archivoslocales con el repositorio remoto. Duración: 1. Responsable: Desarrollador. Entregable: Repositorio de GitHub actualizado con los últimos cambios delproyecto.19.

Despliegue de la base de datos en Clever Cloud Descripción: Configurar y alojar la base de datos del sistema en CleverCloud para garantizar su disponibilidad en un entorno de producción. Incluyela creación de una instancia de base de datos, la configuración decredenciales y la importación de los datos necesarios. Duración: 2. Responsable: Administrador de base de datos (DBA). Entregable: Base de datos funcional y accesible en Clever Cloud.

pg. 3920. Despliegue del backend (aplicación de Django) en Railway / Descripción: Configurar y publicar el backend del sistema web, desarrolladocon Django, en la plataforma Railway. Incluye ajustar configuraciones paraproducción, establecer variables de entorno (como SECRET\_KEY y base dedatos), y realizar el despliegue para que el sistema esté disponible en línea. / Duración: 3. / Responsable:

Desarrollador Backend. / Entregable: Aplicación de Django desplegada y accesible en Railway, con laURL pública para su uso. 21. Creación de las páginas en Wix / Descripción: Diseñar y desarrollar las páginas del sitio web utilizando laplataforma Wix. Esto incluye definir la estructura del sitio, personalizar eldiseño visual (colores, tipografía, imágenes), y agregar funcionalidadesinteractivas mediante Wix Code.

Además, se integran datos dinámicos através de APIs para mostrar contenido actualizado en las páginas. / Duración: 4. / Responsable:

Desarrollador Frontend. / Entregable: Páginas completas y funcionales en Wix, con diseñopersonalizado e integración de datos. 22.

Desarrollo de Controladores y Programación / Descripción: Programar la lógica de negocio del sistema, asegurando lacorrecta integración con la base de datos y la interfaz de usuario. / Duración: 5. / Responsable: Desarrollador backend. / Entregable: Código fuente de los controladores.

pg. 4023. Desarrollo de la Interfaz de Usuario 
Descripción: Diseñar y desarrollar las interfaces gráficas con las queinteractuarán los usuarios. 
Duración: 5. 
Responsable: Desarrollador frontend. 
Entregable: Interfaz de usuario funcional. 
Ambiente de Pruebas 
Descripción: Configurar el entorno de pruebas donde se evaluará elrendimiento y la seguridad del sistema. 
Duración: 2. 
Responsable: Tester/QA. 
Entregable: Entorno de pruebas. 
Descripción: Probar cada componente individual del sistema para 
asegurarsede que cumplen con los requisitos. 
Duración: 5. 
Responsable: Tester/QA. 
Entregable: Reporte de pruebas unitarias. 
Pruebas de Errores del Software 
Descripción: Detectar y corregir errores en el código antes de la integraciónfinal del sistema. 
Duración: 4. 
Responsable: Desarrollador. 
Entregable: Reporte de errores corregidos.

pg. 4127. Pruebas de Conexión con la Base de Datos / Descripción: Verificar que el sistema se conecta correctamente a la base dedatos y realiza las consultas necesarias. / Duración: 3. / Responsable: Tester/QA. / Entregable: Reporte de pruebas de conexión.28. Pruebas de

Aceptación de Usuario 
Descripción: Realizar pruebas con los usuarios finales para asegurarse deque el sistema cumple con sus expectativas. 
Duración: 5. 
Responsable: Tester/QA.Entregable: Reporte de pruebas de aceptación29. Planificación 
Descripción: Definir el cronograma de trabajo y las dependencias entreactividades. 
Duración: 3. 
Responsable: Project Manager. 
Entregable: Cronograma del proyecto. 
30. 
Finalizar Documentación del Software Preliminar 
Descripción: Completar la documentación inicial del 
software, incluyendoespecificaciones técnicas y manuales de usuario. 
Duración: 3. 
Responsable: Documentador técnico. 
Entregable: 
Documentación preliminar.

pg. 4231. Reportes de Pruebas Descripción: Documentar los resultados de todas las pruebas realizadas durante el desarrollo, detallando errores encontrados y su resolución. Duración: 3. Responsable: Tester/QA. Entregable: Reporte final de pruebas.32. Ambiente de Entrega Descripción: Preparar el entorno de producción donde se implementará elsistema para su uso final. Duración: 3. Responsable: Ingeniero de sistemas. Entregable: Ambiente de entrega configurado.33. Entrega de Documentación y Producto Descripción: Hacer la entrega formal del sistema terminado y toda ladocumentación relacionada al cliente. Duración: 2. Responsable: Project Manager. Entregable: Documentación completa y sistema entregado.

pg. 438. MATRIZ DE PRECEDENCIAN.oActividadDescripción De LaActividad Precedencia Dependencias1 Levantamiento deRequerimientos Ninguna Ninguna2 Estudios deFactibilidad 1Requiere los requerimientos levantados 3 Análisis del Sistema Propuesto 1, 2Requiere ellevantamiento derequerimientos y estudiode factibilidad 4 Requerimientos Funcionales 1, 3 Necesita los requerimientos y análisis del sistema 5 Casos De Usos 4 Requiere los requerimientos funcionales 6 Diseño Funcional 4, 5 Basado en los casos deuso y requerimientos funcionales 7 Casos De Prueba 5, 6 Dependiente del diseño funcional y casos de uso 8 Diseño del Sistema 6 Basado en el diseño funcional

pg. 44N.oActividadDescripción De LaActividad Precedencia Dependencias Parquitectura Preliminar 8 Requiere el diseño delsistema 10 Diseño Técnico 9 Basado en la arquitectura preliminar 11 Creación De La BaseDe Datos (SQL.) 9, 10 Necesita el diseño técnico y arquitectura preliminar 12 Diagramas UML QueDocumentan El Proyecto 9, 10 Dependiente del diseño técnico y arquitectura preliminar 13 Despliegue del proyecto a GitHub 9, 10, 11 Requiere el diseño técnico y base de datos 14 Convertir código a Python en model. py 9, 11, 12 Necesita los diagramas UML y diseño técnico 15 Creación del archivoserializar. py 14 Basado en la conversión de código 16 Creación del archivoviews. py 14, 15 Depende del model. py y serializar. py 17 Creación del archivourl. py 16 Requiere el archivoviews. py 18 Re-subir proyecto a GitHub / gitup commit 13, 17 Necesita el despliegue previo y archivos creados 19 Despliegue de la basede datos en Clever Cloud 11, 18 Basado en la base dedatos creada y código subido

pg. 45N.oActividadDescripción De LaActividad Precedencia Dependencias20 Despliegue delbackend en Railway 18, 19Depende del códigosubido y base de datosen la nube21 Creación de laspáginas en Wix 20 Necesita el backenddesplegado22Desarrollo decontroladores yprogramación14, 15, 16, 17 Basado en la estructuradel código23 Desarrollo de lainterfaz de usuario 21, 22 Depende de las páginasde Wix y programación24 Ambiente de pruebas 22, 23 Necesita controladores einterfaz finalizados25 Pruebas Unitarias 24 Requiere el ambiente depruebas26 Pruebas de erroresDel Software 24 Necesita el ambiente depruebas27Pruebas De ConexiónCon La Base DeDatos24 Requiere el ambiente depruebas28Pruebas DeAceptación DeUsuario25, 26, 27 Basado en todas laspruebas previas29 Planificación 1Depende dellevantamiento inicial derequerimientos

pg. 46N.oActividadDescripción De LaActividad Precedencia Dependencias30FinalizarDocumentación DelSoftware Preliminar28 Requiere las pruebas deaceptación31 Reportes De Pruebas 28 Basado en las pruebas deaceptación32 Ambiente De Entrega 30, 31Necesita ladocumentación final yreportes33Entrega DeDocumentación YProducto32 Depende del ambiente deentrega completadoTabla 12 Matriz de Precedencia

pg. 479. MATRIZ DE TIEMPO. (TIEMPO EN HORAS)N.oActividad Descripción O (Óptimo) M (Medio) P (Pésimo)T (Estándar) (T= (O + 4M + P) /6)1LevantamientodeRequerimientos 2 4 7 4.22 Estudios deFactibilidad 1 2 3 2.03Análisis delSistemaPropuesto 3 5 7 5.04

RequerimientosFuncionales 2 5 6 4.75 Casos De Usos 1 3 7 3.36 Diseño Funcional 3 5 8 5.27 Casos DePrueba 2 3 5 3.28 Diseño delSistema

pg. 48N.oActividad DescripciónO(Óptimo)M(Medio)P(Pésimo)T(Estándar)(T = (O + 4M+ P) / 6)11Creación De LaBase De Datos(SQL.)2 5 8 5.012Diagramas UMLQue DocumentanEl Proyecto1 3 5 3.013Despliegue delproyecto aGitHub1 1 2 1.214Convertir códigoa Python enmodel.py3 4 7 4.315Creación delarchivoserializar.py5 7 9 7.016 Creación delarchivo views.py 4 9 12 8.717 Creación delarchivo url.py 1 1 3 1.318Re-subirproyecto aGitHub / gitupcommit1 1 4 1.5

pg. 49N.oActividad DescripciónO(Óptimo)M(Medio) P (Pésimo)T(Estándar)(T = (O +4M + P) /6)19Despliegue de labase de datos enClever Cloud1 1 2 1.220Despliegue delbackend enRailway4 9 15 9.221 Creación de laspáginas en Wix 5 6 18 7.822Desarrollo decontroladores yprogramación6 8 10 8.023Desarrollo de lainterfaz deusuario4 5 11 5.824 Ambiente depruebas 2 5 8 5.025 PruebasUnitarias 4 5 10 5.726Pruebas deerrores DelSoftware3 5 6 4.827Pruebas DeConexión Con LaBase De Datos4 5 7 5.2 pg. 50N.oActividad DescripciónO(Óptimo)M(Medio)P(Pésimo)T(Estándar)(T = (O + 4M+ P) / 6)28Pruebas DeAceptación DeUsuario2 5 5 4.529 Planificación 3 5 10 5.530FinalizarDocumentaciónDel SoftwarePreliminar4 5 20 7.331 Reportes DePruebas 4 5 8 5.332 Ambiente DeEntrega 2 5 7 4.833Entrega DeDocumentaciónY Producto1 5 6 4.5Tabla 13 Matriz de Tiempo

pg. 5110. MATRIZ DE INFORMACIÓN. (TIEMPO EN HORAS)N.o Actividad Descripción Secuencia Tiempo1 Levantamiento deRequerimientos Ninguna 3.82 Estudios de Factibilidad 1 1.83 Análisis del SistemaPropuesto 1, 2 54 RequerimientosFuncionales 1, 3 4.65 Casos De Usos 4 3.36 Diseño Funcional 4, 5 5.17 Casos De Prueba 5, 6 3.18 Diseño del Sistema 6 7.89 Arquitectura Preliminar 8 4.810 Diseño Técnico 9 5.11 Creación De La Base DeDatos (SQL.) 9, 10 512 Diagramas UML QueDocumentan El Proyecto 9, 10 313 Despliegue del proyecto aGitHub 9, 10, 11 1.114 Convertir código a Pythonen model.py 9, 11, 12 4.3

pg. 52N.o Actividad Descripción Secuencia Tiempo15 Creación del archivoserializar.py 14 716 Creación del archivoviews.py 14, 15 8.617 Creación del archivourl.py 16 1.318 Re-subir proyecto aGitHub / gitup commit 13, 17 1.519 Despliegue de la base dedatos en Clever Cloud 11, 18 1.120 Despliegue del backenden Railway 18, 19 9.121 Creación de las páginasen Wix 20 7.822Desarrollo decontroladores yprogramación14, 15, 16, 17 823 Desarrollo de la interfazde usuario 21, 22 5.824 Ambiente de pruebas 22, 23 525 Pruebas Unitarias 24 5.626 Pruebas de errores DelSoftware 24 4.827 Pruebas De ConexiónCon La Base De Datos 24 5.1

pg. 53N.o Actividad Descripción Secuencia Tiempo28 Pruebas De AceptaciónDe Usuario 25, 26, 27 4.529 Planificación 1 5.530 Finalizar DocumentaciónDel Software Preliminar 28 7.331 Reportes De Pruebas 28 5.332 Ambiente De Entrega 30, 31 4.833Entrega DeDocumentación YProducto32 4.5Tabla 14 Matriz de Información

pg. 5411. MATRIZ DE COSTOS.N.o Actividad Descripción Costo (RD\$) Tiempo (horas)1 Levantamiento deRequerimientos 20,000 4.22 Estudios deFactibilidad10,0002.03 Análisis del SistemaPropuesto 15,000 5.04 RequerimientosFuncionales 15,000 4.75 Casos De Usos 10,000 3.36 Diseño Funcional 25,000 5.27 Casos De Prueba 15,000 3.28 Diseño del Sistema 25,000 7.89 ArquitecturaPreliminar 30,000 4.810 Diseño Técnico 25,000 5.011 Creación De La BaseDe Datos (SQL.) 20,000 5.012Diagramas UML QueDocumentan ElProyecto 20,000 3.013 Despliegue delproyecto a GitHub 8,000 1.2

pg. 55N.o Actividad Descripción Costo (RD\$) Tiempo (horas)14 Convertir código aPython en model.py 5,000 4.315 Creación del archivoserializar.py 5,000 7.016 Creación del archivoviews.py 10,000 8.717 Creación del archivourl.py 8,000 1.318Re-subir proyecto aGitHub / gitupcommit5,000 1.519Despliegue de labase de datos enClever Cloud25,000 1.220 Despliegue delbackend en Railway 20,000 9.221 Creación de laspáginas en Wix 25,000 7.822Desarrollo decontroladores yprogramación20,000 8.023 Desarrollo de lainterfaz de usuario 30,000 5.824 Ambiente de pruebas 10,000 5.025 Pruebas Unitarias 15,000 5.7

pg. 56N.o Actividad Descripción Costo (RD\$) Tiempo (horas)26 Pruebas de erroresDel Software 10,000 4.827Pruebas DeConexión Con LaBase De Datos10,000 5.228Pruebas DeAceptación DeUsuario20,000 4.529 Planificación 10,000 5.530FinalizarDocumentación DelSoftware Preliminar10,000 7.331 Reportes De Pruebas 10,000 5.332 Ambiente De Entrega 10,000 4.833Entrega DeDocumentación YProducto15,000 4.5Total 511,000 161.8Tabla 15 Matriz de Costo44 Este precio es igual al Costos de Desarrollo.

pg. 5712. MATRIZ DE RIESGO.Riesgo Probabilidad(%) Impacto Respuesta al RiesgoRetrasos en eldesarrollo debidoa cambios en

losrequerimientos 30 Alto: Afecta plazosy recursos. Establecer un procesoformal de gestión decambios que incluyarevisiones mensuales yaprobación de cambiossignificativos. Fallas en laintegración delbackend (Django) con el frontend (Wix) 20 Alto: Puede afectarla funcionalidad completa. Realizar pruebas deintegración continuas durante el desarrollo, conpruebas unitarias previas acada integración. Falta deexperiencia en eluso de Wix 40 Moderado: Baja eficiencia y problemas técnicos. Organizar capacitaciones y workshops iniciales para elequipo de desarrollo, proporcionando do cumentación y recursos de aprendiza je. Errores en la basede datos MySQL 25 Alto: Pérdida dedatos importantes y posibles interrupciones. Implementar un sistema derespaldo automático diario y realizar auditorías regulares de la base de datos. Sobrecarga delservidor web 15 Alto: Inaccesibilidad dels istema durantepicos de tráfico. Utilizar servidores escalables en la nube, optimizar elcódigo para un mejorrendimiento y monitorizar eltráfico en tiempo real.

pg. 58Riesgo Probabilidad(%) Impacto Respuesta al RiesgoProblemas deseguridadcibernética (SSL)10Alto: Pérdida deconfianza delusuario y violaciónde datos.Implementar medidas deseguridad robustas, como laencriptación de datos yrealizar pruebas depenetración periódicas.Falta decomunicaciónentre los equiposde desarrollo35Moderado:Retrasos y erroresde coordinación.Fomentar una cultura decomunicación abierta, conreuniones diarias breves y uncanal de comunicación (porejemplo, Slack) para resolverdudas rápidamente.Sobreestimaciónde los recursostécnicos25Moderado:Aumento en loscostos y posiblesretrasos.Realizar una planificaciónrealista basada encapacidades previas y hacerun seguimiento continuo deluso de recursos.Desacuerdos conel cliente sobrelasfuncionalidades30Alto: Reversiones yretrabajo queafectan elcronograma.Documentar todos losrequisitos y cambiosacordados, y realizarrevisiones periódicas con elcliente para asegurarse deque las expectativas esténalineadas.Problemas en laspruebas deusuario20Moderado:Necesidad deajustes que podríanretrasar ellanzamiento.Involucrar a los usuariosfinales desde el inicio paraobtener retroalimentaciónconstante y realizar pruebasen fases tempranas deldesarrollo.

pg. 59Riesgo Probabilidad(%) Impacto Respuesta al RiesgoCambios en elentornotecnológico15Moderado: Puederequeriradaptaciones en elproyecto. Mantenerse actualizadosobre las tendenciastecnológicas y ser flexiblepara realizar ajustes en laarquitectura y herramientasutilizadas. Problemas definanciamiento 20Alto: Puede llevar ala paralización delproyecto. Preparar un plan definanciamiento claro yconsiderar fuentesalternativas definanciamiento en caso denecesidad. Tabla 16 Matriz de Riesgo 13. DESCRIPCIÓN DE LAS LIMITACIONES DE RECURSOS.1. Recursos Humanos ✓ Especialización Técnica: La necesidad de conocimientos específicos entecnologías como Django, Wix puede limitar el número de candidatosdisponibles. Si no se encuentran desarrolladores adecuados, el proyectopodría retrasarse o necesitar ajustes en su alcance. ✓ Colaboración en Equipo: Si los miembros del equipo no están familiarizadosentre sí o con las herramientas de gestión de proyectos, esto puede generarconflictos y falta de comunicación, afectando el rendimiento del equipo. pg. 602. Recursos Económicos / Ajustes Presupuestarios: Cambios en las condiciones económicas puedenobligar a recortes en el presupuesto, lo que podría comprometer la calidad delsoftware o limitar la capacitación del personal. ✓ Incertidumbre Financiera: Dependencia de aportes externos (como fondosgubernamentales o donaciones) puede crear incertidumbre en el flujo de cajadel proyecto, haciendo difícil planificar gastos a largo plazo.3. Recursos Materiales ✓ Obsolescencia de Equipos: Equipos de cómputo obsoletos pueden ralentizarel desarrollo y pruebas del sistema, afectando la productividad del equipo. ✓ Limitaciones de Software: Herramientas de desarrollo que no cumplen conlos estándares requeridos o que son difíciles de usar pueden generarfrustración y disminuir la eficiencia del equipo.4. Recursos de Tiempo ✓ Planificación de Tiempos: La subestimación del tiempo necesario paracompletar ciertas actividades puede llevar a una acumulación de tareas y a unretraso general en el cronograma. Pruebas Inadecuadas: Si el tiempo dedicado a las pruebas es insuficiente, pueden surgir errores en producción que impacten la experiencia del usuario yrequieran correcciones costosas y tardadas.5. Recursos Externos ✓ Dependencia de Proveedores: La dependencia de proveedores paraservicios como hosting o licencias de software puede crear vulnerabilidades siestos proveedores no cumplen con sus compromisos. ✓ Cambios Normativos: Cambios en la legislación o normativas que regulan laeducación superior o el uso de datos personales pueden requerirmodificaciones en el sistema, afectando el cronograma y los costos.

pg. 61Recomendaciones para Mitigar Limitaciones1. Planificación Proactiva: Realizar un análisis detallado de las competenciasdel equipo y

asignar tareas según habilidades, además de planificar lacapacitación necesaria desde el inicio. 2. Presupuesto Flexible: Incorporar márgenes de maniobra en el presupuestopara imprevistos y cambios de alcance.3. Revisiones Periódicas: Implementar reuniones regulares para evaluar elprogreso, ajustar tiempos y recursos según sea necesario.4. Establecimiento de Contingencias: Tener planes de contingencia para losriesgos identificados, como la búsqueda de proveedores alternativos o laposibilidad de reubicar tareas dentro del equipo en caso de ausencias. 5. Documentación Clara: Mantener una documentación clara y accesible paratodos los miembros del equipo, lo que facilitará la comunicación y lacontinuidad del trabajo, incluso ante cambios de personal.14. CALENDARIO DE EJECUCIÓN DE PROYECTOS.14.1 Fechas relevantesCódigos Entregables Fecha De EntregaE01 Documentación delproyecto (backend)30 sept 2024E02 Diseño de la interfaz deusuario (HTML)03 oct 2024E03 Código fuente delsistema08 oct 2024E04 Implementación delsistema11 oct 2024E05 Pruebas y validación 13 oct 2024E06 Manual de usuario 21 oct 2024Tabla 17 Calendario de Ejecución pg. 6215. DIAGRAMA DE GANTTActividades Fecha deinicio Fecha definalización Asignado Estado Proyecto Práctico 08.09.2024 29.09.2024 Jose A. Terminado1. Introducción 08/09/2024 09.09.2024 Deivi Terminado2. Descripción General Del Proyecto 09.09.2024 10.09.2024 Yonger Terminado3. Equipo De Trabajo 10.09.2024 11.09.2024 Jose A. Terminado4. Descripción De Los Aspectos Técnicos 11.09.2024 12.09.2024 Deivi Terminado5. Presupuesto 12.09.2024 13.09.2024 Yonger Terminado6. Lista De Actividades. 13.09.2024 15.09.2024 Jose A. Terminado7. Descripción De Actividades 15.09.2024 17.09.2024 Deivi Terminado8. Matriz De Precedencia 16.09.2024 18.09.2024 Yonger Terminado 9. Matriz De Tiempo. (tiempo en horas) 17.09.2024 19.09.2024 Jose A. Terminado 10. Matriz De Información 18.09.2024 20.09.2024 Deivi Terminado11. Matriz De Costos 19.09.2024 21.09.2024 Yonger Terminado12. Matriz De Riesgo 20.09.2024 22.09.2024 Jose A. Terminado13. Descripción De Las Limitaciones De Recursos 21.09.2024 23.09.2024 Deivi Terminado14. Calendario De Ejecución De Proyectos. 22.09.2024 24.09.2024 Yonger Terminado15. Diagrama de Gantt 23.09.2024 25.09.2024 Jose A. Terminado16. Herramientas De Seguimiento Y Control 24.09.2024 26.09.2024 Deivi Terminado17. Conclusiones Y Recomendaciones 25.09.2024 27.09.2024 Yonger Terminado16. Bibliografía Y Referencias Electrónicas 26.09.2024 28.09.2024 Jose A. Terminado19. Anexos 27.09.2024 29.09.2024 Deivi TerminadoLevantamiento de Requerimientos 30.09.2024 04.10.2024 Jose A. TerminadoEstudios de Factibilidad 30.09.2024 02.10.2024 Deivi TerminadoAnálisis del Sistema Propuesto 30.09.2024 05.10.2024 Yonger TerminadoRequerimientos Funcionales 05.10.2024 10.10.2024 Jose A. TerminadoCasos De Usos 03.10.2024 06.10.2024 Deivi TerminadoDiseño Funcional 06.10.2024 11.10.2024 Yonger TerminadoCasos De Prueba 11.10.2024 14.10.2024 Jose A. TerminadoDiseño del Sistema 07.10.2024 15.10.2024 Deivi TerminadoArquitectura Preliminar 12.10.2024 17.10.2024 Yonger TerminadoDiseño Técnico 15.10.2024 20.10.2024 Jose A. TerminadoCreación De La Base De Datos (SQL.) 15.10.2024 20.10.2024 Terminado Diagramas UML Que Documentan El Proyecto 16.10.2024 19.10.2024 Deivi TerminadoDespliegue del proyecto a gitup 18.10.2024 19.10.2024 Yonger TerminadoConvertir código a python en model.py 21.10.2024 25.10.2024 Jose A. TerminadoCreación del archivo serializar.py 20.10.2024 27.10.2024 Deivi TerminadoCreación del archivo views.py 20.10.2024 29.10.2024 Yonger TerminadoCreación del archivo url.py 26.10.2024 27.10.2024 Jose A. TerminadoRe-subir proyecto a gitup /gitup and commit 30.10.2024 01.11.2024 Yonger TerminadoDespliegue de la base de datos en clever cloud 28.10.2024 29.10.2024 Jose A. TerminadoDespliegue del backend (aplicación de Django):Railway 29.10.2024 08.11.2024 Deivi TerminadoCreación de las páginas en wix 02.11.2024 08.11.2024 Yonger TerminadoDesarrollo de controladores y programación 30.10.2024 08.11.2024 Jose A. Terminado pg. 63Desarrollo de la interfaz de usuario 09.11.2024 14.12.2024Deivi, Jose y Yonger Terminado Ambiente de pruebas 10.11.2024 15.12.2024Pruebas Unitarias 11.11.2024 16.12.2024Pruebas de errores Del Software 12.11.2024 17.12.2024Pruebas De Conexión Con La Base De Datos 13.11.2024 18.12.2024 Pruebas De Aceptación De Usuario 14.11.2024 19.12.2024 Planificación 15.11.2024 20.12.2024 Finalizar Documentación Del Software Preliminar 16.11.2024 21.12.2024Reportes De Pruebas 17.11.2024 22.12.2024Ambiente De Entrega 18.11.2024 23.12.2024Entrega De Documentación Y Producto 19.11.2024 24.12.2024 pg. 64Figura 2.0 Diagrama GANTT pg. 6516. HERRAMIENTAS DE SEGUIMIENTO Y CONTROLEn el desarrollo y ejecución del Sistema Web de Gestión de Pagos e Ingresos de

pg. 6516. HERRAMIENTAS DE SEGUIMIENTO Y CONTROLEn el desarrollo y ejecución del Sistema Web de Gestión de Pagos e Ingresos de laUnidad de Educación Continua, se emplearán diversas herramientas para asegurarun seguimiento efectivo del progreso del proyecto y un

control adecuado de losrecursos. A continuación, se describen las principales herramientas que se utilizarán:Herramientas de Gestión de Proyectos > Monday: Se utilizará para la planificación y gestión de tareas. Monday permiteorganizar actividades, asignar tareas a miembros del equipo y establecerplazos, facilitando el seguimiento del avance del proyecto en tiempo real. Herramientas de Comunicación > Slack: Se implementará como plataforma de comunicación interna. Slackfacilita la comunicación en tiempo real, la creación de canales temáticos y laintegración con otras herramientas, lo que ayuda a mantener al equipoconectado y al tanto de las actualizaciones del proyecto. > Microsoft Teams: Además de la mensajería, Teams permitirá realizarvideoconferencias y compartir archivos, lo que es fundamental para reunionesy colaboraciones remotas. Herramientas de Documentación > Google Drive: Para el almacenamiento y colaboración en documentos, hojasde cálculo y presentaciones. Google Drive permite a los miembros del equipotrabajar en documentos simultáneamente, facilitando la actualización y revisiónde la documentación del proyecto. > Dropbox: Se usará para la gestión y almacenamiento de documentos delproyecto, permitiendo el acceso fácil y seguro a la documentación por parte delequipo.

pg. 66Herramientas de Desarrollo > GitHub: Para el control de versiones del código fuente, GitHub permitirágestionar cambios en el código, colaborar en el desarrollo y llevar un registrode las modificaciones realizadas por cada miembro del equipo. > Postman: Para realizar pruebas de las API desarrolladas, Postman facilitarála verificación del correcto funcionamiento de los endpoints del sistema. Herramientas de Monitoreo > UptimeRobot: Se utilizará para monitorizar el estado y el rendimiento delsistema una vez implementado. UptimeRobot enviará alertas en caso decaídas del servicio, lo que permitirá una respuesta rápida para minimizar eltiempo de inactividad. Herramientas de Pruebas > Selenium: Para realizar pruebas automatizadas de la interfaz de usuario. Selenium ayudará a garantizar que las funcionalidades del sistema secomporten correctamente en diferentes navegadores y dispositivos. > Jira: En caso de que se requiera un enfoque más robusto para la gestión deincidencias y pruebas, Jira puede ser utilizado para rastrear errores y gestionarel ciclo de vida de los mismos.

pg. 6717. CONCLUSIONES Y RECOMENDACIONES17.1 ConclusionesEl diseño y desarrollo del sistema web en Python para la gestión de pagos e ingresosde cursos y diplomados en la Unidad de Educación Continuada de la Facultad deCiencias Económicas y Sociales (FCES) representa un avance significativo en laoptimización de los procesos administrativos de la institución. A través de laimplementación de tecnologías modernas como Django para el backend y Wix parael frontend, se ha logrado una solución eficiente y accesible que mejora la experienciadel usuario y fortalece la gestión interna. El sistema desarrollado cumple con los objetivos planteados, abarcando aspectosclave como el diseño de la arquitectura, la implementación de un módulo de registroy seguimiento de pagos, la generación de reportes financieros y la integración denotificaciones. Además, se llevaron a cabo pruebas de rendimiento y seguridad paragarantizar su estabilidad y funcionalidad. El desarrollo del proyecto fue un proceso colaborativo en el que el equipo enfrentó ysuperó diversos retos técnicos y organizacionales mediante una adecuadaplanificación, distribución de tareas y comunicación constante. La adopción de estesistema no solo permite automatizar la gestión financiera, sino que también optimizael uso de recursos y reduce errores administrativos.

pg. 6817.2 Recomendaciones1) Ampliación de funcionalidades: Se sugiere considerar la integración demódulos adicionales, como un sistema de retroalimentación de los estudiantesy un portal de seguimiento académico. Esto permitiría mejorar la interacción ysatisfacción de los usuarios, expandiendo el impacto del sistema más allá dela gestión financiera.2) Capacitación del personal administrativo: Para maximizar elaprovechamiento del sistema, es recomendable ofrecer capacitacionesperiódicas al personal encargado de la gestión de pagos e ingresos, asegurando un uso eficiente de la plataforma.3) Mantenimiento y actualizaciones: Se recomienda establecer un plan demantenimiento continuo para garantizar el correcto funcionamiento delsistema, actualizar la seguridad y mejorar su rendimiento a largo plazo.4) Optimización de la experiencia del usuario: Se pueden realizar mejoras enla interfaz gráfica y la navegabilidad de la plataforma para hacerla aún másintuitiva y accesible.5) Monitoreo y análisis de datos: Implementar herramientas de análisis dedatos para evaluar la eficiencia del sistema y su impacto en la gestiónadministrativa, permitiendo ajustes y mejoras basadas en información objetiva. Estamos

seguro de que estas recomendaciones van a evolucionar el sistema yconsolidarse como una solución integral para la gestión de pagos e ingresos en laFCES, asegurando su sostenibilidad y efectividad a largo plazo.

pg. 6916. BIBLIOGRAFÍA Y REFERENCIAS ELECTRÓNICAS Holovaty, A., & Willison, S. (21 de julio de 2005). django Framework.

Obtenidode django: https://www.djangoproject.com/ Maffulli, S. (8 de febrero de 1998). Open-Source Initiative. Obtenido de

OpenSourceInitiative: https://opensource.org/licenses Mateu, C. (2006). Desarrollo de aplicaciones web. En C. Mateu, Desarrollo

deaplicaciones web (pág. 378). Barcelona: Eureca Media, SL. Otto, M., & Thornton, J. (19 de agosto de 2011). Bootstrap. Obtenido

deBootstrap: https://getbootstrap.com/

pg. 7019. ANEXOSA1 (Guía de entrevistas - Preguntas formuladas al cliente)1. ¿Qué tipo de información quieren generar con el sistema?Estados financieros, evaluación del tiempo, mejorar los pagos.2. ¿Tienen algún plazo o fecha límite para implementar la solución?- 30 días, pero el tiempo real sería de dos meses.3. ¿Cómo se diferencian los pagos con distintos cursos o diplomados?- Son iguales en proceso, pero los cursos tienen diferentes tarifas, mientras que losdiplomados están estandarizados.4. ¿Cómo se manejan los pagos parciales?- El solicitante de reembolso debe presentar una carta firmada. Se puedeautomatizar el proceso.5. ¿Quién es responsable del control de los pagos?- Un equipo de personal administrativo se encarga del seguimiento de los pagos.6. ¿Cómo se confirman las inscripciones de un estudiante una vez realizado elpago?- A través de un correo institucional conectado al área financiera.7. ¿Necesitan enviar recordatorios de pago pendiente o notificaciones deingresos?- Sí, es necesario.8. ¿Qué métodos de pago utilizan?- Todos los métodos, incluyendo transferencias, Paypal y pagos manuales.9. ¿Tienen algún requisito de seguridad específico para los datos de pago einformación de los estudiantes?

pg. 71- Sí, es importante asegurar la privacidad de la información.10. ¿Cómo registran a los estudiantes en el curso?- Es un proceso manual y en línea, se busca automatizar más.11. ¿Cuáles son los datos más importantes que necesitan recopilar de losalumnos?- Nombre, cédula, correo, número de teléfono y referencias.12. ¿Cómo administran los contenidos y materiales de los cursos?- No quieren que los materiales sean accesibles desde la plataforma para evitarproblemas de ética.13. ¿Qué método de pago utilizan?- Utilizan manual, Paypal y transferencias.14. ¿Tienen personal que puede administrar el sistema una vez esté enfuncionamiento?- Sí, hay personal capacitado para administrar el sistema.15. ¿Con qué frecuencia necesitan actualizar la información sobre los cursoso pagos?- Diario.16. ¿Tienen planes de crecimiento en términos de cantidad de cursos odiplomados?- Sí, siempre están creciendo.17. ¿Cómo miden el éxito de esta nueva solución?- A través del aumento de ingresos y la agilización de los trabajos de los profesores.18. ¿Qué aspecto consideran más importante en el sistema?- Que sea fácil de usar y altamente personalizado.19. ¿Hay alguna otra plataforma o sistema que hayan visto que les gustaríatomar como referencia?

pg. 72- Se mencionó que se admiraban sistemas existentes, pero no se específicó uno.20. ¿Qué tipo de reportes necesitan generar regularmente?- Reportes de ingresos, pagos pendientes y asistencia a cursos.21. ¿Cómo planean realizar el seguimiento de las interacciones de losestudiantes con la plataforma?- Se utilizarán herramientas de análisis para monitorear la actividad de los usuarios.22. ¿Existen algunos datos que deben ser obligatorios para el registro de losestudiantes?- Sí, el nombre, cédula y correo son obligatorios.23. ¿Tienen alguna preferencia sobre la tecnología a utilizar para el desarrollodel sistema?- No se mencionó una preferencia específica, pero buscan algo robusto y fácil demanejar.24. ¿Qué tipo de soporte técnico necesitarán una vez que el sistema esté enfuncionamiento?- Necesitarán soporte continuo y un manual de usuario.25. ¿Qué nivel de acceso tendrá cada usuario en la plataforma?- Habrá diferentes niveles de acceso, dependiendo del rol (administrativo, docente, estudiante).26. ¿Cómo piensan integrar el sistema con otras herramientas o plataformasque ya usan?- Buscarán formas de integrar mediante APIs o importaciones/exportaciones dedatos.27. ¿Tienen alguna preferencia sobre el diseño o la interfaz de usuario?- Quieren un diseño intuitivo y amigable para todos los usuarios.28. ¿Qué medidas de seguridad tienen en mente para proteger los datos?- Implementar encriptación y accesos controlados.

pg. 7329. ¿Tienen algún plan para capacitar al personal en el uso del sistema?- Sí, se realizarán capacitaciones una vez esté implementado. 30. ¿Cómo planean manejar las quejas o problemas que puedan surgir con eluso del sistema?- Se establecerá un canal de

atención al usuario para resolver inquietudes.31. ¿Qué características consideran esenciales para la gestión de pagos?- Facilidad de uso, múltiples métodos de pago y reportes automáticos.32. ¿Qué tan importante es la personalización de la experiencia del usuario?- Muy importante, ya que desean que cada usuario tenga una experiencia adaptadaa sus necesidades.33. ¿Tienen en mente realizar pruebas piloto antes del lanzamiento delsistema?- Sí, realizarán pruebas piloto con un grupo reducido de usuarios.34. ¿Qué tipo de feedback esperan obtener de los usuarios una vez que elsistema esté en uso?- Esperan comentarios sobre la facilidad de uso, funcionalidad y eficiencia.35. ¿Tienen algún plan para actualizar el sistema en el futuro?- Sí, desean implementar actualizaciones periódicas basadas en el feedbackrecibido.36. ¿Qué tipo de contenido esperan que los estudiantes puedan acceder através del sistema?- Material de cursos, recursos adicionales y foros de discusión.37. ¿Cómo piensan evaluar el éxito del sistema después de suimplementación?- A través de métricas de uso, satisfacción del usuario y rendimiento en la gestión depagos.

pg. 7438. ¿Qué rol jugarán los docentes en la plataforma?- Los docentes tendrán la capacidad de cargar contenido, gestionar inscripciones yevaluar a los estudiantes.39. ¿Planean incorporar alguna función de gamificación en el sistema?- Están considerando elementos de gamificación para aumentar la motivación de losestudiantes.40. ¿Cómo manejarán la comunicación entre estudiantes y docentes?- A través de un sistema de mensajería interna y foros de discusión.41. ¿Tienen un plazo definido para el desarrollo e implementación delsistema?- Sí, buscan implementarlo en un plazo de seis meses.42. ¿Qué recursos adicionales consideran necesarios para el éxito delproyecto?- Recursos humanos, financieros y técnicos para asegurar un desarrollo fluido.43. ¿Hay alguna regulación específica que deban cumplir con el sistema?- Deben cumplir con las normativas de protección de datos y accesibilidad.44. ¿Qué plan de contingencia tienen en caso de problemas técnicos duranteel uso del sistema?- Se establecerá un protocolo de emergencia para resolver problemas rápidamente.45. ¿Qué aspectos consideran críticos para la formación de los usuarios?- La capacitación en el uso del sistema y la comprensión de las políticas de datos.46. ¿Cómo piensan promover el sistema entre los estudiantes y el personal?- A través de campañas de información y sesiones de demostración.47. ¿Tienen alguna visión a largo plazo para el sistema?- Expandirlo para incluir más funciones y cursos en el futuro.48. ¿Cómo planean recoger datos para la mejora continua del sistema?

pg. 75- Mediante encuestas y análisis de uso regular.49. ¿Qué elementos de diseño consideran esenciales para la interfaz?- Simplicidad, accesibilidad y un diseño responsive.50. ¿Cómo se asegurará la calidad del contenido que se sube al sistema?- Se establecerán procesos de revisión para garantizar la calidad del material.

pg. 76A2.0 (Factura cotización de equipos Mínimo)

pg. 77A2.1 (Factura cotización de equipos Media)

pg. 78A2.2 (Factura cotización de equipos Máximo)

pg. 79A2.3 (Factura cotización de equipos Recomendada)

pg. 80A2.4 (Factura cotización de equipos Recomendado)

pg. 81A2.5 (Factura cotización de equipos Mínimo)

pg. 82A3.0 (-Base de datos conectada)-- Crear la base de datos para la gestión de pagos y cursosCREATE DATABASE gestion\_pagos\_cursos;-- Usar la base de datos recién creadaUSE gestion\_pagos\_cursos;-- Tabla Usuarios: almacena información básica de los usuarios del sistemaCREATE TABLE Usuarios (id INT AUTO\_INCREMENT PRIMARY KEY, -- Identificador único para cadausuarionombre\_completo VARCHAR (255) NOT NULL, -- Nombre completo del usuariocedula VARCHAR (20) NOT NULL UNIQUE, -- Cédula única del usuariocorreo VARCHAR (255) NOT NULL UNIQUE, -- Correo electrónico único delusuariotelefono VARCHAR (15), -- Número de teléfono (opcional)celular VARCHAR (15), -- Número de celular (opcional)direccion TEXT, -- Dirección del usuario (opcional)contraseña VARCHAR (255) NOT NULL, -- Contraseña encriptada del usuariofecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Fecha decreación del registroestado ENUM ('activo', 'inactivo') DEFAULT 'activo', -- Estado del usuarioUNIQUE (cedula, correo) -- Asegura que la combinación de cédula y correo seaúnica

pg. 83);-- Tabla Estudiantes: hereda de Usuarios y almacena información específica deestudiantesCREATE TABLE Estudiantes (id INT

```
PRIMARY KEY, -- Identificador único (heredado de Usuarios) matricula VARCHAR (20) NOT NULL UNIQUE, -- Matrícula única del
estudiantefecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- Fecha deregistro del estudianteFOREIGN KEY (id) REFERENCES
Usuarios(id) ON DELETE CASCADE --Relación con la tabla Usuarios); -- Tabla Docentes: hereda de Usuarios y almacena información
específica dedocentesCREATE TABLE Docentes (id INT PRIMARY KEY, -- Identificador único (heredado de Usuarios)especialidad
VARCHAR (255) NOT NULL, -- Especialidad del docentefecha_contratacion DATE NOT NULL, -- Fecha de contratación del
docentefacultad VARCHAR (255) NOT NULL, -- Facultad a la que pertenece el docenteescuela VARCHAR (255) NOT NULL, -- Escuela
dentro de la facultadcampus VARCHAR (255) NOT NULL, -- Campus donde trabaja el docenteestatus ENUM ('activo', 'jubilado') NOT
NULL DEFAULT 'activo', -- Estado laboraldel docentecodigo ENUM ('invitado', 'oficial') NOT NULL, -- Tipo de código del
docenteFOREIGN KEY (id) REFERENCES Usuarios(id) ON DELETE CASCADE --Relación con la tabla Usuarios
pg. 84);-- Tabla Administrativos: hereda de Usuarios y tiene permisos de accesoCREATE TABLE Administrativos (id INT PRIMARY KEY, --
Identificador único (heredado de Usuarios)departamento VARCHAR (255) NOT NULL, -- Departamento al que pertenece
eladministrativocargo VARCHAR (255) NOT NULL, -- Cargo del administrativofecha_ingreso DATE NOT NULL, -- Fecha de ingreso del
administrativoacceso ENUM ('solo_ver', 'ver_agregar', 'superusuario') NOT NULL DEFAULT'solo_ver', -- Nivel de acceso del
administrativo FOREIGN KEY (id) REFERENCES Usuarios(id) ON DELETE CASCADE --Relación con la tabla Usuarios);-- Tabla Cursos:
almacena información sobre los cursos disponiblesCREATE TABLE Cursos (id INT AUTO_INCREMENT PRIMARY KEY, -- Identificador único
para cada cursonombre VARCHAR (255) NOT NULL, -- Nombre del cursodescripcion TEXT, -- Descripción del cursotipo ENUM ('curso',
'diplomado') NOT NULL, -- Tipo de curso (curso o diplomado)tarifa DECIMAL (10, 2) NOT NULL, -- Tarifa del cursoestado ENUM ('activo',
'inactivo') DEFAULT'activo', -- Estado del cursofecha inicio DATE NOT NULL, -- Fecha de inicio del cursofecha fin DATE NOT NULL, --
Fecha de finalización del cursocapacidad INT DEFAULT 0, -- Capacidad máxima del cursodocente_id INT, -- Relación con Docentes
pg. 85modulos INT NOT NULL, -- Número de módulos del cursohoras INT NOT NULL, -- Total de horas del cursocodigo VARCHAR(20)
NOT NULL, -- Código del cursoprofesor VARCHAR(255) NOT NULL, -- Nombre del profesor (alternativa adocente_id)facultad
VARCHAR(255) NOT NULL, -- Facultad a la que pertenece el cursotelefono VARCHAR(15) NOT NULL, -- Teléfono de contacto para el
cursoimagen_url VARCHAR(255), -- URL de la imagen del cursoFOREIGN KEY (docente_id) REFERENCES Docentes(id) ON DELETE
SETNULL -- Relación con la tabla Docentes);-- Tabla Inscripciones: almacena las inscripciones de los estudiantes en los cursosCREATE
TABLE Inscripciones (id INT AUTO INCREMENT PRIMARY KEY, -- Identificador único para cadainscripciónestudiante id INT, -- Relación
con la tabla Estudiantescurso_id INT, -- Relación con la tabla Cursosfecha_inscripcion TIMESTAMP DEFAULT CURRENT_TIMESTAMP, --
Fecha deinscripciónestado ENUM('inscrito', 'pendiente') NOT NULL, -- Estado de la inscripciónFOREIGN KEY (estudiante_id) REFERENCES
Estudiantes(id) ON DELETECASCADE, -- Relación con la tabla EstudiantesFOREIGN KEY (curso_id) REFERENCES Cursos(id) ON DELETE
CASCADE -- Relación con la tabla Cursos);
pg. 86-- Tabla Pagos: almacena información sobre los pagos realizados por losestudiantesCREATE TABLE Pagos (id INT
AUTO_INCREMENT PRIMARY KEY, -- Identificador único para cada pagoinscripcion_id INT, -- Relación con la tabla
Inscripcionesmetodo_pago ENUM('transferencia', 'Paypal', 'manual') NOT NULL, -- Método depago utilizadomonto DECIMAL(10, 2) NOT
NULL, -- Monto del pagoestado_pago ENUM('completado', 'pendiente', 'parcial') NOT NULL, -- Estado delpagofecha_pago TIMESTAMP
DEFAULT CURRENT TIMESTAMP, -- Fecha derealización del pagofecha vencimiento DATE, -- Fecha de vencimiento del pagoFOREIGN
KEY (inscripcion_id) REFERENCES Inscripciones(id) ON DELETECASCADE -- Relación con la tabla Inscripciones);-- Tabla Historial de
Pagos: para auditar cambios en los estados de pagoCREATE TABLE Historial_Pagos (id INT AUTO_INCREMENT PRIMARY KEY, --
Identificador único para cadaregistro de historialpago_id INT, -- Relación con la tabla Pagosestado_pago_anterior ENUM('completado',
'pendiente', 'parcial'), -- Estadoanterior del pagoestado_pago_nuevo ENUM('completado', 'pendiente', 'parcial'), -- Nuevo estadodel pago
pg. 87fecha_cambio TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- Fecha delcambio de estadocomentario TEXT, -- Comentario sobre el
```

cambio de estado FOREIGN KEY (pago\_id) REFERENCES Pagos(id) ON DELETE CASCADE --Relación con la tabla Pagos);-- Tabla

Reembolsos: almacena información sobre solicitudes de reembolsoCREATE TABLE Reembolsos (id INT AUTO\_INCREMENT PRIMARY KEY, --- Identificador único para cadasolicitud de reembolsousuario\_id INT, -- Relación con la tabla Usuariospago\_id INT, -- Relación con la tabla Pagosmotivo TEXT NOT NULL, -- Motivo de la solicitud de reembolsoestado ENUM('pendiente', 'aprobado', 'rechazado') NOT NULL, -- Estado de lasolicitudfecha\_solicitud TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Fecha desolicitud de reembolsofecha\_resolucion TIMESTAMP NULL, -- Fecha de resolución del reembolso (siaplica)FOREIGN KEY (usuario\_id) REFERENCES Usuarios(id) ON DELETECASCADE, -- Relación con la tabla UsuariosFOREIGN KEY (pago\_id) REFERENCES Pagos(id) ON DELETE CASCADE -- Relación con la tabla Pagos);

pg. 88-- Tabla Notificaciones: almacena notificaciones enviadas a los usuariosCREATE TABLE Notificaciones (id INT AUTO\_INCREMENT PRIMARY KEY, -- Identificador único para cadanotificaciónusuario\_id INT, -- Relación con la tabla Usuariostipo\_notificacion ENUM('recordatorio\_pago', 'ingreso\_registrado') NOT NULL, --Tipo de notificaciónmensaje TEXT NOT NULL, -- Mensaje de la notificaciónfecha\_envio TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Fecha deenvío de la notificaciónestado ENUM('enviado', 'pendiente') DEFAULT 'pendiente', -- Estado de lanotificaciónFOREIGN KEY (usuario\_id) REFERENCES Usuarios(id) ON DELETE CASCADE-- Relación con la tabla Usuarios); -- Tabla Reportes: almacena reportes generados por el sistemaCREATE TABLE Reportes (id INT AUTO\_INCREMENT PRIMARY KEY, -- Identificador único para cadareportetipo\_reporte ENUM('ingresos', 'pagos\_pendientes', 'asistencia') NOT NULL, --Tipo de reportefecha\_generacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, -- Fecha degeneración del reportedescripcion TEXT NOT NULL -- Descripción del reporte);

pg. 89A4.0 (-Diccionario de Datos)Python: Es un lenguaje de programación ampliamente utilizado en las aplicacionesweb, el desarrollo de software, la ciencia de datos y el machine learning (ML).Optimizar: La optimización de software es el proceso de mejorar el rendimiento, laeficiencia y la calidad de las aplicaciones de software.Sistema web: Son todos aquellos softwares que son utilizados accediendo a travésde un servidor web, sin necesidad de tener una aplicación en el escritorio.Dispositivo: Un dispositivo informático es un ordenador que proporciona uno omás servicios.Django: Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo-vista-controlador.Backend: Son los datos y la infraestructura que permiten que la aplicación funcione.Almacena y procesa los datos de las aplicaciones para los usuarios.FronTend: El desarrollo web front-end se refiere a la práctica de construir y diseñarla interfaz de usuario de un sitio web o aplicación.MySQL: MySQL es un sistema de gestión de bases de datos relacional desarrolladobajo licencia dualInterfaz de usuario: Es el punto de interacción y comunicación humano-computadoraen un dispositivoWix: Es una compañía SaaS que ofrece servicios de desarrollo web basados en lanubeDiseñador UI/UX: Es el profesional que se encarga de supervisar la experiencia deusuario, la UX (User Experience).QA: El aseguramiento de la calidad es el conjunto de actividades planificadas ysistemáticas aplicadas en un sistema de gestión de la calidad para que los requisitos de calidad de un producto o servicio sean satisfechos.Servidor web: Un servidor web o servidor HTTP es un programa informático queprocesa una aplicación del lado del servidor, realizando conexiones bidireccionales ounidireccionales y síncronas o asíncronas con el cliente y generando o cediendo unarespuesta en cualquier lenguaje o aplicación del lado del cliente.

pg. 90Hardware: El hardware, equipo o soporte físico en informática se refiere a las partesfísicas, tangibles, de un sistema informático, sus componentes eléctricos, electrónicosy electromecánicos. Software: Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistemade computación. Licencia de software: Es un contrato entre el licenciante y el licenciatario delprograma informático, para utilizarlo cumpliendo una serie de términos y condiciones. La nube: Es una metáfora para designar una red mundial de servidores remotos quefunciona como un único ecosistema, normalmente asociada a Internet. Mantenimiento: Todas las acciones que tienen como objetivo preservar un artículoo restaurarlo a un estado en el cual pueda llevar a cabo alguna función requerida. Automatización: Consiste en usar la tecnología para realizar tareas con muy pocaintervención humana. Autenticación: La autenticación o autentificación es el acto o proceso de confirmarque algo es quien dice ser. A la parte que se identifica se le llama probador. UptimeRobo: Monitorea sitios web, servidores, API y servicios en línea para verificarsu

disponibilidad y disponibilidadArquitectura del sistema: Es el modelo conceptual que define la estructura, elcomportamiento y más vistas de un sistema. Navegador web: Es un programa que permite ver la información que contiene unapágina web. Roles: Es un conjunto de permisos que se comparten entre uno o más usuarios. Encriptar: Es el proceso de codificar un mensaje o información de modo tal que sololos individuos autorizados sean capaces de acceder a esta. Código fuente: Es un archivo o conjunto de archivos, que contienen instruccionesconcretas, escritas en un lenguaje de programación, que posteriormente compilanuno o varios programas. Código QR: Es la evolución del código de barras. Es un módulo para almacenarinformación en una matriz de datos o en un código de barras bidimensional Backups: Es una copia de los datos de una computadora que se toma y se almacenaen otro lugar para que pueda usarse para restaurar el original después de un eventode pérdida de datos.

pg. 91Api: Es una pieza de código que permite a dos aplicaciones comunicarse entre sípara compartir información y funcionalidades.Linux Ubuntu: Ubuntu es una distribución GNU/Linux basada en Debian GNU/Linux, que incluye principalmente software libre y de código abierto.Ghz: El gigahercio es un múltiplo de la unidad de medida de frecuencia hercio yequivale a 109 Hz. Por lo tanto, tiene un período de oscilación de 1 nanosegundo. Frameworks: Es un conjunto estandarizado de conceptos, prácticas y criterios paraenfocar un tipo de problemática particular que sirve como referencia, para enfrentar yresolver nuevos problemas de índole similar.XAMPP: Es un paquete de software libre, que consiste principalmente en el sistemade gestión de bases de datos MySQL, el servidor web Apache y los intérpretes paralenquajes de script PHP y Perl. Apache 2.4: Es un servidor web de código abierto, el más popular de los queencontramos en el mundo de la web. Como todo servidor web, maneja solicitudesHTTPNginx: Es un servidor web/Proxy inverso ligero de alto rendimiento y un proxy paraprotocolos de correo electrónico. Rsync: Es una aplicación libre para sistemas de tipo Unix y Microsoft Windows queofrece transmisión eficiente de datos incrementales, que opera también con datoscomprimidos y cifrados. Uncomplicated Firewall ò Firewall: Es un dispositivo de red, que puede ser software(host-based) o un dispositivo de hardware, utilizado para monitorear el tráfico quepasa a través de él, con el fin de bloquear o permitir paquetes, basándose en unconjunto de reglas y capacidades. Endpoints apis: Son los puntos en los que el cliente y el servidor se comunican, loque permite que dos aplicaciones compartan recursos. Http: Es el protocolo de comunicación que permite las transferencias de informacióna través de archivos en la World Wide Web.SSL/TLS: Seguridad de la capa de transporte y su antecesor Secure Sockets Layerson protocolos criptográficos, que proporcionan comunicaciones seguras por una red, comúnmente Internet. Cross-Site Scripting (XSS).: Es un tipo de ataque informático que permite a un actorde amenazas ejecutar código malicioso en el navegador de otro usuario.

pg. 92Certificado SSL: Es un tipo de documento digital firmado por una entidad llamadaAutoridad de certificación que asocia una clave pública a unos datos que representanla identidad de una entidad que posee la clave privada asociada a dicha clave pública. Diagrama UML: El lenguaje unificado de modelado es el lenguaje de modelado desistemas de software más conocido y utilizado en la actualidad, respaldado por elObject Management Group. Diagrama de Gantt: Es una herramienta gráfica cuyo objetivo es exponer el tiempode dedicación previsto para diferentes tareas o actividades a lo largo de un tiempototal determinado.

pg. 93A5.0 (-Cotización)DESARROLLO DE TECNOLOGIA CORPORATIVARNC:132355131Cotización:DISEÑO Y DESARROLLO DE UN SISTEMA WEB ENPYTHON PARA LA GESTIÓN DEPAGOS E INGRESOS DECURSOS Y DIPLOMADOS EN LA UNIDAD DEEDUCACIÓNCONTINUADA DE LA FACULTAD DE CIENCIASECONÓMICAS YSOCIALES (FCES)Cliente: Licda. MaribelDescripción General del ProyectoDesarrollo de un sistema web en Python para la gestión de pagos e ingresos de cursos ydiplomados en la unidad de educación continuada de la Facultad De CienciasEconómicas Y Sociales (FCES), con el objetivo de mejorar el registro, administración depagos, y otras transacciones estudiantiles. El sistema permitirá la gestión de pagos, generación de reportes financieros, y contará con acceso seguro para administradoresy personal autorizado de la universidad.Tecnologías:-Backend: Django (Python)-Frontend: Wix-Base de datos: MySQL- Diseño UI/UX:- Control de calidad: Testeos funcionales y de usabilidadPersonal Involucrado1. Desarrolladores Backend (Django) - 22. Desarrolladores Frontend (Wix) - 23. Diseñador UI/UX - 14. Tester/QA - 15. Administrador de Base de Datos (DBA) - 1 pg. 94Alcance del Proyecto1. Módulo de Administración de Usuarios: Registro, autenticación y gestión de roles.2. Módulo de Pagos:

Generación de pagos, pasarela de pagos en línea, gestión depagos pendientes. 3. Módulo de Reportes: Reportes financieros, exportación de datos a formatos PDF/Excel. 4. Interfaz de Usuario: Diseño intuitivo y accesible para personal universitario administrativo. 5. Seguridad: Configuración de permisos, cifrado de datos sensibles, y protección dedatos personales. Presupuesto Detallado Cargo Cantidad Tasa(USD/Hora) Horas Estimadas Subtotal (USD) Desarrollador Backend (Django) 2 18 83 1494,000 Desarrollador Frontend (Wix) 2 15 91 2,730,000 Diseñador UI/UX 1 15 91 1,365,000 Tester/QA 1 12 17 204,000 DBA 1 16 17 272,000 Total Estimado 7 76 299 6,061.00 Detalle de Entregables 1. Análisis y Diseño del Sistema- Análisis de requerimientos: 2 semanas- Diseño UI/UX y especificaciones técnicas: 2 semanas 2. Desarrollo y Codificación- Backend (APIs, lógica de negocio, seguridad): 6 semanas- Frontend (Interfaz en Wix, integración con backend): 5 semanas- Base de Datos (Esquema y optimización): 2 semanas 3. Pruebas y Control de Calidad- Pruebas unitarias y funcionales: 3 semanas- Pruebas de integración y aceptación: 2 semanas pg. 954. Implementación y Entrenamiento- Implementación en entorno de producción: 1 semana- Entrenamiento del personal universitario: 1 semana Términos y Condiciones- Forma de Pago: 50% al inicio, 25% a la entrega de los módulos funcionales, 25% ala entrega final.- Tiempo de Desarrollo: Aproximadamente 5 meses desde la fecha de inicio delproyecto.- Mantenimiento: Un mes de soporte técnico postlanzamiento, incluido en el costo. Para cualquier aclaración, no dude en contactarnos. Nos complace la oportunidadde colaborar con Licda. Maribel Lorenzo Linares y contribuir al mejoramiento desu sistema administrativo. Revisado por: Ing. Carlos García Cell: 809-657-1414 Fecha: 08/10/2024

## **SCAN SETTINGS**

These features were chosen to create this report

Omitsettings	Repositories	
References:	Off	
Quotes:	Off	
Citations:	Off	
Titles:	Off	
HTML Templates:	Off	
Table Of Contents:	Off	
Code Comments:	Off	

**Plagiarism Detection Settings** 

Security Measures	
Safe Search:	Off
Hide Sensitive Data:	Off
Character Manipulation:	Off
Similarity Level	
Identical:	On
Minor Changes:	On

Results Calibration	
Focused Results	Sensitivity: 3