Términos de referencia

Especificaciones técnicas para el diseño y desarrollo de una aplicación web que funcione como sistema de gestión para el ordenamiento ecológico territorial del estado de Yucatán

Tabla de contenido

1 Nombre de la prestación de servicios. 2
2 Introducción 2
3 Objetivos 3
3.1 Objetivo general 3
3.2 Objetivos específicos. 3
3.2.1 Diseñar y desarrollar el módulo de administración 3
3.2.2 Diseñar y desarrollar el módulo de consultas y descargas 3
3.2.3 Diseñar y desarrollar el módulo de actualización de recursos 6
3.2.4 Diseñar y desarrollar el módulo de automatización de reportes 6
3.2.5 Desarrollar el diseño gráfico de la aplicación 7
3.2.6 Atender las observaciones surjan en el periodo de pruebas 8
3.2.7 Atender los problemas técnicos derivados del uso del sistema 8
4 Productos y entregables 8
5 Requisitos de la propuesta 9
5.1 Plan de trabajo y metodologías 9
5.2 Capacidades del equipo de trabajo 9
5.3 Experiencia en proyectos similares 9
Índice de anexos: 10

1 Nombre de la prestación de servicios.

Diseño y desarrollo de una aplicación web, que funcione como sistema gestión para el ordenamiento ecológico del territorio del estado de Yucatán.

2 Introducción

El ordenamiento ecológico es un instrumento de política ambiental cuyo objetivo es planear la distribución de los usos de suelo en el territorio. Éste establece una zonificación y regulaciones para el desarrollo de las actividades productivas. Estas regulaciones se utilizan como base para aprobar o rechazar proyectos que requieren aprobación en materia ambiental. Con el fin de transparentar, eficientar y potenciar las actividades relacionadas a este instrumento de política ambiental, tanto de ciudadanos como de autoridades, se requiere un sistema de gestión para el ordenamiento territorial del estado de Yucatán.

El sistema de gestión se concibe como un sistema geográfico de soporte de decisiones basado en la integración de plataformas de geo-visualización, manejo y análisis de datos y generación de reportes. El propósito del sistema de gestión es mejorar tangiblemente las capacidades instaladas en el gobierno de Yucatán para el manejo, análisis y visualización de información que requiere para la formulación, implementación y monitoreo del ordenamiento ecológico, así como su articulación con otros instrumentos de planeación y gestión. Asimismo, debe facilitar el acceso público y uso de la información del Programa de ordenamiento ecológico del estado de Yucatán (POETY) para promover la participación corresponsable de los actores sociales en este proceso de política pública.

El sistema se compondrá de dos subsistemas funcionales: (1) la bitácora ambiental y (2) un subsistema de información y comunicación interna. La bitácora ambiental será de acceso público, mientras que sistema de información y comunicación interna será un módulo de acceso exclusivo para los funcionarios de la SDS. Las necesidades de la Secretaría de Desarrollo Sustentable (SDS) en relación con el sistema de la gestión del POETY se describen en el anexo I.

El propósito de la bitácora ambiental es generar un registro público del proceso del ordenamiento ecológico, en cumplimiento con las obligaciones legales (Reglamento de la Ley General del Equilibrio Ecológico y la Protección al Ambiente en Materia de Ordenamiento Ecológico).

El propósito del subsistema de información y comunicación interna es que exista una plataforma al interior de la SDS, mediante el cual todas las áreas compartan información (documentos, bases de datos, capas geográficas) que sirvan como insumo para que los funcionarios lleven a cabo los procedimientos relacionados con la ejecución del POETY de manera

más eficiente. Éste deberá tener una interfase con el Ecosistema Georreferenciado del Estado de Yucatán (EGEY) y la plataforma para la gestión de trámites que maneja la Subsecretaría de Tecnologías de la Información y Comunicaciones (STIC) del Gobierno de Yucatán.

3 Objetivos

3.1 Objetivo general

Diseñar y desarrollar una aplicación web, que funcione como sistema gestión para el ordenamiento ecológico del territorio del estado de Yucatán.

3.2 Objetivos específicos.

Los objetivos específicos deben apegarse a los casos de uso y los componentes de software descritos en el anexo II. Siguiendo los lineamientos de los anexos III y IV, cumpliendo los requisitos de los anexos V, VI, VII, y tomando en cuenta las recomendaciones del anexo VIII.

3.2.1 Diseñar y desarrollar el módulo de administración

Dado que algunas de las funcionalidades solo serán accesibles para usuarios internos de la SDS, es necesario contar con un módulo de administración de usuarios que permita la creación, edición y eliminación de usuarios con sus respectivas contraseñas.

El sistema se desarrolla en el contexto de la actualización del Programa de Ordenamiento Ecológico Territorial del Estado de Yucatán (POETY), sin embargo, el sistema debe tener la capacidad de agregar otros ordenamientos ecológicos, por ejemplo, Programa de Ordenamiento Ecológico del Territorio Costero del Estado de Yucatán (POETCY) o Programa de Ordenamiento Ecológico Local (POEL), por lo que en este módulo se incluye también la capacidad de crear, modificar y eliminar ordenamientos ecológicos.

3.2.2 Diseñar y desarrollar el módulo de consultas y descargas

Este módulo debe proveer los mecanismos para consultar y en su caso descargar toda la información asociada a cada ordenamiento ecológico que se agregue en el sistema. Cada ordenamiento de dividirá en las siguientes secciones:

- Introducción

- Comité de Ordenamiento ecológico
- Talleres de participación pública
- Estudio técnico
- Programa de Ordenamiento Ecológico
- Sistema de monitoreo y evaluación

Este módulo permitirá acceder al contenido de cada una de estas secciones. El contenido estará compuesto de documentos y mapas interactivos, en la Tabla 1 se enlista el contenido para cada una de las secciones.

Tabla 1. Contenido de las secciones.

Sección	Objetivo	Contenido
Introducción	Difundir información general sobre el POETY	• Texto en la página web que describe datos relevantes del POETY: objetivo y uso, fecha de suscripción y partes firmantes del convenio, etc. • Hipervínculo con el convenio de coordinación (formato pdf.)
Comité de Ordenamiento ecológico	Difundir documentación relacionada con la operación del Comité de Ordenamiento ecológico	• Texto en la página web que describe el objetivo del comité, su conformación y fecha de instalación Subsecciones **1. Instalación** Hipervínculos con el acta de instalación y el reglamento interno (formato pdf.) **2. Sesiones del comité de ordenamiento ecológico** 2.1. Órgano Ejecutivo 2.2. Órgano Técnico 2.3. Plenarias Ordenar sesiones por fecha y para cada sesión incluir hipervínculos con los siguientes documentos (formato pdf.): o orden del día o lista de asistencia o acta de la sesión o material complementario: fotos, presentaciones, documentos, etc.
Talleres de participación pública	Difundir documentación relacionada con los talleres que se desarrollan en el proceso de actualización del POETY	Ordenar talleres por fecha y para cada uno incluir hipervínculos con los siguientes documentos (formato pdf.): o orden del día o lista de asistencia o minuta del taller o material complementario: fotos, presentaciones, documentos, etc.

Estudio técnico	Difundir los informes que entrega el equipo técnico de las etapas del estudio técnico	Hipervínculos con los informes de cada etapa (formato zip. o rar.) • caracterización • diagnóstico • pronóstico • propuesta • Mapas de las etapas de caracterización, diagnóstico, y pronóstico • Mapa de aptitud para cada sector productivo • Mapas de grupos de aptitud • Mapa de conflictos espaciales • Mapas de áreas de atención prioritaria
Programa de Ordenamiento Ecológico	Difundir el programa de ordenamiento ecológico y permitir la consulta de los criterios de regulación por UGA y por polígono.	• Hipervínculo con la publicación oficial del POETY (formato pdf.) • Liga a la plataforma de geovisualización (EGEY) • Capas de información temática (formato shapefile o raster) • Bancos de datos asociados a las capas de información temática (formato excel) • Secciones del informe del estudio técnico que describe cómo se aplicaron los análisis asociados a las capas generadas (por ejemplo, los mapas de aptitud, de conflictos, etc.) (formato pdf.) • Bancos de datos y textos que van a conformar las fichas de las UGAs • Mapa de UGAs • Fichas técnicas de cada UGA (ver el anexo II) • Reporte de criterios de regulación por polígono (ver el anexo II)
Monitoreo y evaluación	Difundir los indicadores que permitan evaluar la implementación del POETY	• Texto en la página web que describe el sistema de indicadores para el monitoreo y evaluación. • Geovisualización de información sobre los indicadores. Complementada con tablas y gráficas que faciliten la comunicación del significado de los indicadores y las tendencias. • Liga a la plataforma de geovisualización (EGEY) y de gestión de trámites de SCTIC) • Liga a plataformas de otros programas de la SDS (monitoreo de calidad de agua, etc.) • Hipervínculo con reportes de evaluación del POETY con base en el análisis de los indicadores

3.2.3 Diseñar y desarrollar el módulo de actualización de recursos

El módulo de actualización de recursos permitirá a usuarios internos de la SDS ingresar, modificar y eliminar los documentos que se describen en el objetivo específico 3.2.3. Dado que las capas geográficas y los datos de proyectos que requieren autorización en materia ambiental se hospedarán en las bases de datos del EGEY y de la STIC respectivamente y que la actualización de estos datos requiere de un trámite oficial por parte de la SDS, este tipo de actualización no será parte del sistema. Sin embargo, como parte del diseño y desarrollo de este módulo, se deben incluir las especificaciones de las tablas y capas geográficas para su integración y compatibilidad con los módulos de consulta y descarga y de automatización de reportes.

3.2.4 Diseñar y desarrollar el módulo de automatización de reportes

Este módulo se encargará de generar en tiempo real tres componentes de contenido dinámico:

- Fichas técnicas de las Unidades de Gestión Ambiental.
- Reportes de criterios de regulación por polígono.
- Reportes de monitoreo y evaluación del desempeño del programa de ordenamiento ecológico.

La información necesaria para generar estos documentos se irá acumulando en las bases de datos del EGEY donde se hospedarán las capas geográficas y del STIC donde se hospedarán los datos de proyectos que requieren autorización en materia de impacto ambiental.

3.2.4.1 Fichas descriptivas de las Unidades de Gestión Ambiental

Las unidades de gestión ambiental son polígonos en el territorio dentro de los cuales aplican una serie de restricciones llamados criterios de regulación. Los objetivos de las fichas descriptivas son:

- Mostrar la ubicación de la UGA.
- Proveer una descripción general del territorio que ocupa la UGA.
- Listar los criterios de regulación que aplican en la UGA.
- Proveer información sobre los impactos acumulados.

La ubicación de la UGA y los criterios de regulación son estáticos en el sentido de que una vez establecidos no cambiaran hasta que haya una actualización del ordenamiento ecológico. Sin embargo, los impactos acumulativos se actualizarán conforme se vayan aprobando los proyectos con autorización en materia de impacto ambiental. El sistema deberá conectarse a

la base de datos de proyectos aprobados, para extraer los datos necesarios para incluir una evaluación de los impactos acumulativos en la ficha descriptiva de cada UGA. Esta consulta a la base de datos de proyectos aprobados debe hacerse de manera periódica para que la información esté siempre actualizada.

3.2.4.2 Reportes de criterios de regulación por polígono

Este es un reporte en el que el usuario provee un polígono de interés, y el sistema construye de forma automatizada un reporte con los criterios de regulación que aplican en el polígono. Dado que un polígono puede estar contenido en más de una UGA el reporte debe contener la información de cada una de las intersecciones, y los criterios de regulación que aplican en cada una de ellas. En este sentido estos reportes son documentos únicos para cada consulta y se deberán construir en tiempo real. El usuario podrá proveer el polígono de interés mediante la digitalización interactiva dentro del mapa en el sistema, o bien, proporcionar un archivo geográfico en formato shapefile, kmz, o geojson. Y el sistema generará un reporte en formato pdf. Dado que uno de los casos de uso accesibles para usuarios internos del sistema es consultar el historial de reportes por polígono que se generen. El sistema debe guardar en su base de datos, los polígonos que proporcionen los usuarios y los documentos que se generen en consecuencia.

3.2.4.3 Reportes de monitoreo y evaluación del desempeño del programa de ordenamiento ecológico

Estos reportes se ensamblarán de manera periódica cada seis meses y deben contener información sobre el desempeño del programa de ordenamiento ecológico. La información que servirá de insumo para generar estos reportes será la base de datos de proyectos con autorización en materia de impacto ambiental, la base de datos de condicionantes por proyecto, datos de campo medidos y reportados por personal de la SDS, una serie de indicadores de desempeño, e información de bases de datos externas o capas geográficas externas (ejemplo: INEGI, CONAFOR, CONABIO). En estos reportes se ubicarán espacialmente los proyectos aprobados, se integrarán gráficas y mapas para comunicar el significado de los indicadores y las tendencias, y se dará seguimiento a proyectos estratégicos federales y estatales prioritarios, generando indicadores de sostenibilidad.

3.2.5 Desarrollar el diseño gráfico de la aplicación

El diseño de la interfaz de usuario debe ser intuitiva y fácil de usar, además de, ser compatible con la imagen institucional del gobierno del estado de Yucatán que se describe en el anexo III. Y seguir lineamientos en el formato, de las hojas de estilo y los demás códigos propios del diseño, que se describen en el anexo IV.

3.2.6 Atender las observaciones surjan en el periodo de pruebas

Una vez desarrollado el sistema en una primera versión funcional, debe someterse a un periodo de pruebas, de cuatro semanas, en el cual se atenderán las observaciones por parte del personal de LANCIS y la SDS de Yucatán. El objetivo de este periodo es ajustar el código, en caso de ser necesario, tomando en cuenta la retroalimentación de los usuarios principales del sistema.

3.2.7 Atender los problemas técnicos derivados del uso del sistema

A partir del lanzamiento al público del sistema de gestión, el prestador del servicio se comprometerá a atender las fallas técnicas del sistema por periodo de un año. Para lo cual debe establecerse un mecanismo de mesa de servicio que sirva de canal de comunicación entre personal de la STIC y los desarrolladores del sistema. La finalidad de esta etapa de soporte técnico es corregir las posibles fallas del sistema derivadas de su uso público.

4 Productos y entregables

1. Plan de trabajo definitivo. Esto será un documento que establezca el plan de trabajo con cronograma detallado para desarrollar los objetivos específicos 3.2.1-3.2.5 en un término de 6 meses. Este documento se entregará en un periodo de dos semanas a partir de la firma del contrato de prestación de servicios.
2. Repositorio creado en GitHub o GitLab con acceso a personal de LANCIS.
3. Diseño gráfico de todas las vistas de la aplicación. Este será un documento en el que se incluirán las vistas de la aplicación conforme a lo establecido en el objetivo específico 3.2.5.
4. Código con documentación de los cuatro módulos descritos en los objetivos específicos. Los entregables serán el código albergado en el repositorio establecido y una demostración de la funcionalidad desarrollada en un taller con personal del LANCIS y la SDS.
5. Manual de instalación de la aplicación. Será un documento que describa los pasos de configuración del servidor y de instalación de la aplicación necesarios para dejar la plataforma funcional.

6. Manual de uso y administración del sistema. Será un documento que describa como se usa y administra el sistema.
7. Informe de ajustes de la etapa de prueba de cuatro semanas. Este será un documento que contenga la descripción de los ajustes que se lleven a cabo derivados de la retroalimentación por parte del personal de LANCIS y la SDS.
8. Informe de fallas técnicas del primer año. Este será un documento que contenga la descripción las fallas técnicas que surjan en el uso de la aplicación, así como las soluciones a estas.

5 Requisitos de la propuesta

5.1 Plan de trabajo y metodologías

Breve descripción del enfoque y metodologías para la prestación de los servicios descritos anteriormente, indicando los principales factores / problemas a considerar y cómo se abordarán. Así como un cronograma tentativo donde se establezcan los hitos principales.

5.2 Capacidades del equipo de trabajo

CV del personal clave, incluidos expertos en desarrollo de software, bases de datos y manejo de información geográfica.

5.3 Experiencia en proyectos similares

Experiencia de la empresa que demuestre la prestación exitosa de servicios similares al descrito anteriormente.

Índice de anexos:

I. Necesidades del usuario y otros beneficiarios
II. Diseño y arquitectura del sistema de gestión
III. Guía de estilos para el diseño de páginas web
IV. Lineamientos de trabajo web
V. Políticas de bases de datos para terceros
VI. Políticas para bases de datos geoespaciales
VII. Procedimiento de entrega de documentos de base de datos para terceros
VIII. Recomendaciones de seguridad web para considerar en el desarrollo de la página

Anexo I

Necesidades del usuario y otros beneficiarios

Sistema de conocimiento para la gestión del Ordenamiento Ecológico del Territorio (OET) del Estado de Yucatán

4. Proyecto del sistema
4.1 Necesidades del usuario y otros beneficiarios

Juntos transformemos Yucatán
GOBIERNO ESTATAL 2018 · 2024
SDS SECRETARÍA DE DESARROLLO SUSTENTABLE
CONACYT
LANCIS Laboratorio Nacional de Ciencias de la Sostenibilidad

Índice del capítulo 4.1

4.1 Identificación de necesidades del usuario y otros beneficiarios 2

4.1.1 Contenido y funciones 2

4.1.1.1 Bitácora ambiental del POETY 2

4.1.1.2 Sistema de información y comunicación interna 4

4.1.2 Identificación de lineamientos institucionales..... 6

4.1 Identificación de necesidades del usuario y otros beneficiarios

A partir del diagnóstico institucional (sección 1 de este informe) y la consideración de las obligaciones legales en materia de ordenamiento ecológico se identificaron las necesidades de la Secretaría de Desarrollo Sustentable (SDS) en relación con el diseño del sistema de conocimiento y soporte geográfico de decisiones para la gestión del POETY.

En un primer nivel se reconoce la necesidad de que el sistema incluya tanto la bitácora ambiental como un sistema de información y comunicación interna.

El objetivo de la bitácora ambiental es cumplir con las obligaciones legales en materia de ordenamiento ecológico conforme lo establecido en el Reglamento de la LGEEPA en la materia de generar un registro público del proceso. Éste será un módulo abierto a la consulta pública.

El objetivo del sistema de información y comunicación interna es que exista una plataforma al interior de la SDS, mediante el cual todas las áreas compartan información (documentos, bases de datos, capas digitales) que sirvan como insumo para que los funcionarios lleven a cabo los procedimientos relacionados con la ejecución del POETY de manera más eficiente. Éste será un módulo al cuál únicamente tendrán los funcionarios de la SDS, pero que también tendrá una interfase con el sistema estatal de información (Ecosistema de Gestión Territorial-proyecto OCDE).

4.1.1 Contenido y funciones

A continuación, se desglosan los contenidos y funciones para cada uno de los dos componentes:

4.1.1.1 Bitácora ambiental del POETY

Introducción

En esta sección se describe el objetivo del POETY, su aplicación y datos que se consideren relevantes)

- Convenio de coordinación
- Se describe la fecha de suscripción del convenio y se especifican qué autoridades los firmaron
- Se incluye una liga con el convenio en formato pdf. para su consulta

Comité de ordenamiento ecológico

En esta sección se publicarán todos los documentos relacionados con la instalación y operación del comité de ordenamiento ecológico

- Se describe el objetivo del comité y datos sobre su conformación y fecha de instalación
- Se incluyen ligas con el acta de instalación y el reglamento interno del comité de ordenamiento ecológico en formato pdf. para su consulta

- Se incluye una sección de “Sesiones del comité de ordenamiento ecológico” en la que se van a publicar los documentos de cada sesión, incluyendo:
    - Orden del día
    - Lista de asistencia
    - Información complementaria (fotos, presentaciones, documentos)
    - Actas (separar sesiones del Órgano Ejecutivo del Órgano Técnico y Plenarias y ordenar sesiones por fecha)

Talleres de participación pública

En esta sección se incluyen los documentos de cada taller

- Orden del día de talleres de participación pública
- Lista de asistencia de talleres de participación pública
- Actas o minutas de talleres de participación pública

Estudio técnico

En esta sección se incluyen los informes que entrega el equipo técnico de las etapas del estudio técnico

- Agenda ambiental
- Informe o resumen ejecutivo de la etapa de caracterización del estudio técnico
- Informe o resumen ejecutivo de la etapa de diagnóstico del estudio técnico
- Informe o resumen ejecutivo de la etapa de pronóstico del estudio técnico
- Informe o resumen ejecutivo de la etapa de propuesta del estudio técnico

Programa de Ordenamiento Ecológico

- Se incluye una liga con el decreto del programa - publicación oficial – en formato pdf.
- Plataforma de geovisualización (ejemplo ver Versatile http://magrat.mine.nu:8080/Versatile/)

  En esta plataforma los ciudadanos podrán:

    - consultar informes del estudio técnico – asociados a los mapas que se generan (por ejemplo, los mapas de aptitud, de conflictos, etc.)
    - visualizar mapas usados y generados en el estudio técnico
    - consultar bancos de datos utilizadas en el estudio técnico
    - descargar mapas (formato shapefile)
    - introducir polígono de proyecto - con cursor o coordenadas y consultar ficha técnica de unidad de gestión ambiental (UGA) donde se encuentra y que el usuario pueda bajar
        - tabla con todos los atributos de la UGA e incorporar a un visualizador
        - lista de criterio de regulación que le aplican
        - para que pueda utilizar esa información en la elaboración de la MIA o ETJ

Sistema de monitoreo y evaluación

Esta sección se generará una vez que se definan los indicadores como resultado del estudio técnico

- Introducción en la que se explica el sistema de indicadores que se definirá en la actualización del POETY
- Publicación de información sobre los indicadores

  Se utilizarán herramientas (tablas, gráficas y/o coordenadas paralelas asociadas a mapas, etc.) que faciliten la comunicación del significado de los indicadores y las tendencias.

- Visualización de los proyectos autorizados en materia de impacto ambiental
  - ubicar espacialmente los proyectos
  - consultar documentos de resolutivos
  - consultar indicadores de los impactos en el territorio
- Reportes de evaluación del POETY con base en el análisis de los indicadores

4.1.1.2 Sistema de información y comunicación interna

Gestión de trámites

El diseño deberá contemplar los procedimientos de gestión de los siguientes trámites que emite la SDS:

- Factibilidad Urbana Ambiental (FUA)
- Evaluación de la manifestación de impacto ambiental
- Emisión de opinión técnica a la SEMARNAT sobre proyectos que solicitan autorización de cambio de uso de suelo en terrenos forestales.

Los funcionarios podrán llevar a cabo las siguientes acciones:

- Dar de alta proyectos a evaluar:
  - cargar datos del proyecto en un formulario
  - incluir una interfaz con el sistema de intranet que ya tiene la SDS para las FUA o con el sistema de trámites a nivel estatal - Ecosistema de Gestión Territorial
  - introducir polígono de proyecto - con cursor o coordenadas

- Hacer consultas y descarga de información que aplica al proyecto en un machote de opinión técnica prellenado, de las siguientes fuentes:
  - POETY y otros POE: mapas, bases de datos y criterios de regulación ecológica que aplican al proyecto
  - Repositorio de mapas y bases de datos de otras fuentes disponibles en la SDS (ANP, cenotes, PDU, atlas de riesgo, INEGI, etc.

- Interfaz para jalar datos de otras bases de datos, p.e. UNAM, CONABIO, CONAFOR, INECC, etc.
- Catálogo de impactos / condicionantes que se integre con base en la experiencia de todos los evaluadores de la SDS
- Repositorio de proyectos autorizados para analizar impactos acumulativos (éste se tendrá que ir generando una vez que se opere el sistema)

- Generar un repositorio de las opiniones técnicas que servirá como insumo del Sistema de monitoreo y evaluación (*** insumo para sistema de monitoreo)
  - llevar registro de proyectos con opiniones positivas, negativas o que quedaron inconclusas
  - hacer una base de datos con información clave de los proyectos autorizados (por ejemplo: superficie autorizada de cambio de uso de suelo en terrenos con vegetación forestal, volumen de agua utilizado, generación de residuos – información que se obtenga de las MIA) que sirva para generar indicadores de los impactos en el territorio

SDS podría hacer gestión con los gobiernos municipales y la SEMARNAT para evaluar la posibilidad de tener registro de las autorizaciones que ellos emiten.

Manejo del sistema de monitoreo y evaluación

En este módulo se deberá procesar la información que se requiere para alimentar el sistema de monitoreo del POETY que se publique en la bitácora ambiental.

Los funcionarios podrán llevar a cabo las siguientes acciones:

- Integrar la información que se requiere para calcular valores de los indicadores generados en la actualización del POETY. Para ello se deberá contar con las interfases necesarias para que puedan tener acceso a las siguientes fuentes de información:
  - bancos de datos de otros programas de la SDS (estrategia de transversalidad) por ejemplo, con sistema de evaluación de calidad del agua
  - bancos de datos de otras dependencias del gobierno estatal (Ecosistema de Gestión Territorial-proyecto OCDE)
  - bancos de dato de los gobiernos federal y municipales
  - bancos de datos de instituciones académicas

- Dar seguimiento a proyectos estratégicos – federales y estatales prioritarios
  Se podrán generar indicadores de sostenibilidad para poner evaluar estos proyectos

- Dar seguimiento a proyectos autorizados, utilizando la información que se registra como parte de la gestión de trámites***

Con base en esta información se podrán:

- ubicar espacialmente los proyectos y visualizar en un mapa, incluyendo la posibilidad de acceder los resolutivos correspondientes
- generar indicadores de los impactos que tienen los proyectos en el territorio, con base en la información de la manifestación de impacto ambiental
- dar seguimiento de condicionantes (se podría tener una interfaz en la que los promoventes suban reportes de los programas que se establecieron en las condicionantes, por ejemplo, monitoreo de calidad de agua en cenotes, etc.)

4.1.2 Identificación de lineamientos institucionales

Con el fin de conocer los lineamientos de la SDS y la Subsecretaría de Tecnologías de la Información y Comunicaciones (STIC) del Gobierno de Yucatán en materia de informática para considerarlos en el diseño del sistema de conocimiento y soporte geográfico de decisiones para la gestión del POETY, se llevaron a cabo cuatro reuniones virtuales con 28 de abril, 1, 8 y 11 de junio con los siguientes funcionarios:

- Sara Hernilda Aguilar Castillejos, Jefe del Departamento de Ordenamiento Ecológico Territorial, SDS
- Andrés Sierra Gómez, Jefe de Departamento de Geoestadística y Proyectos, SDS
- Carlos Antonio Varguez, Enlace de informática, SDS
- David Mex, Director de Estrategia de Datos, STIC

Como resultado de las reuniones se acordó que el diseño y el desarrollo del sistema debe alinearse con el proyecto de la OCDE que coordina la STIC para desarrollar una plataforma para la gestión de trámites y el Ecosistema Georreferenciado del Estado de Yucatán (EGEY)

En el caso de la SDS el sistema de gestión de trámites automatizará los procedimientos de evaluación de impacto ambiental, factibilidad urbana ambiental y licencia ambiental única. El sistema está acoplado con el EGEY en el que los funcionarios podrán consultar información para hacer sus opiniones y subir la información que se genera en la dependencia.

Considerando este avance, se acordó que el componente de información y comunicación interna que propuso LANCIS para el sistema de conocimiento del POETY, ya está contemplado en el proyecto de la OCDE, en términos del desarrollo de las plataformas.

Para este componente el papel del LANCIS se limitará a apoyar a la SDS en la generación de información que permita fortalecer el sustento técnico de las opiniones. Esto podrá incluir la integración de un catálogo de impactos y condicionantes con base en la experiencia de los evaluadores de proyectos en SDS, la definición de indicadores, el uso de la información de reportes de condicionantes para evaluar pertinencia de condicionantes, entre otros. Esta información podrá integrarse posteriormente al sistema de la STIC. Asimismo, una vez que se actualice el POETY se podrán incorporar en el EGEY los criterios de regulación ecológica e indicadores de cambios en el territorio para evaluar el desempeño del POETY.

Debido a que el componente de la bitácora ambiental no está considerado en los alcances del sistema que desarrolla STIC, se acordó que LANCIS diseñará y desarrollará la bitácora ambiental. Esto incluye, definir

la estructura del sitio web (contenido) y programar la aplicación, considerando las especificaciones técnicas que proporcionó la STIC:

- Guía de estilos para el diseño de páginas web
- Lineamientos de trabajo web
- Políticas de bases de datos para terceros
- Políticas para bases de datos geoespaciales
- Procedimiento de entrega de documentos de base de datos para terceros
- Recomendaciones de seguridad web para considerar en el desarrollo de la página

LANCIS identificará los requerimientos de usabilidad del sistema, de acuerdo con los contenidos de la bitácora ambiental (tamaño de los archivos, número de clic para llegar a la información, visualización en el EGEY y consultas que pueden hacerse y la información que va a obtener el ciudadano).

El enlace de informática de la SDS señaló que la bitácora ambiental actual está obsoleta, en cuanto al diseño y la funcionalidad para cargar información de forma dinámica. Propone que la bitácora ambiental sea una página web dinámica y responsiva (que se pueda consultar desde cualquier dispositivo), que tenga la siguiente estructura:

- un panel de administración con usuarios y contraseñas para que los funcionarios de la SDS puedan ir cargando la información que se va generando en la actualización del POETY y su implementación
- el sistema de información geográfica - una sección en la que se integren al EGEY los insumos cartográficos que se desarrollen en la actualización del POETY para su visualización y para que el ciudadano haga las consultas geoespaciales.

Las bases de datos generadas en la actualización del POETY deberán cumplir los lineamientos de SCTIC y se integrarán al EGEY para que se pueda desplegar la representación visual.

La SDS solicita que la bitácora ambiental sea una página en la que se puedan cargar los demás ordenamientos ecológicos que haya en el estado. Sobre esta propuesta se acordó desarrollar conjuntamente una guía de requisitos para que la SDS pueda incorporar en la plataforma la información de los demás procesos, ya que el alcance de este proyecto es únicamente en relación con el POETY.

Anexo II

Diseño y arquitectura del sistema de gestión

Sistema de conocimiento para la gestión del Ordenamiento Ecológico del Territorio (OET) del Estado de Yucatán

4. Proyecto del sistema

4.2 Diseño y arquitectura del sistema de gestión

Juntos transformemos Yucatán
GOBIERNO ESTATAL 2018 · 2024
SDS SECRETARÍA DE DESARROLLO SUSTENTABLE
CONACYT
LANCIS Laboratorio Nacional de Ciencias de la Sostenibilidad

Índice del capítulo 4.2

4.2 Diseño y arquitectura del sistema de gestión.....2

4.2.1 Especificación de requerimientos..... 2

4.2.1.1 Alcances.....2

4.2.1.2 Objetivos generales 2

4.2.1.3 Requerimientos.....3

4.2.1.4 Beneficios 4

4.2.1.5 Limitaciones 5

4.2.1.6 Casos de uso.....5

4.2.1.7 Diagramas de actividad 37

4.2.2 Arquitectura..... 46

4.2 Diseño y arquitectura del sistema de gestión

Esta sección contiene los requerimientos funcionales y arquitectónicos del diseño del sistema de gestión del POETY. La primera subsección aborda los requerimientos del sistema, es decir describe los procesos y capacidades de los usuarios. Mientras que la segunda subsección describe la arquitectura del sistema, esto es, los componentes de software que lo componen y la forma en la que se integran.

4.2.1 Especificación de requerimientos

Esta subsección documenta los requerimientos funcionales del sistema. El proposito es describir las necesidades del sistema y las actividades y procesos que las atienden.

El sistema será una aplicación web que dará servicio tanto al público en general, como a usuarios internos del sistema. Los usuarios internos del sistema tendrán que autenticarse para acceder a funcionalidades permitidas solo para usuarios internos. Las computadoras desde las cuales los usuarios internos accedan al sistema deben contar con un procesador de al menos 2.4 GHz, y al menos 8 Gb de memoria RAM, se recomienda el uso de pantallas con al menos 1920x1080 de resolución y al menos 27 pulgadas, para una visualización efectiva. Aunque el sistema debe funcionar con cualquier resolución y tamaño de pantalla.

El servidor que aloje el sistema debe contar con aproximadamente 100GB de espacio en disco. La mayoria de los recursos del sistema estarán albergados el las bases de datos del EGEY y del STIC. Sin embargo el registro de los reportes de regulación por polígono se hara directamente en la base de datos del sistema, por esta razón y para contemplar la inclusion de otros ordenamientos, se recomienda reservar 100GB de espacio en disco duro. En cuanto al ancho de banda en el servidor se recomienda al menos 200 Mbps. Lo cual se traduciría en la capacidad de atender 1000 usuarios concurrentes descargando capas de información geográfica con tiempos de descarga de 30 segundos por capa. Finalmente, en cuanto a la memoria RAM del servidor, se requiere un mínimo de 26GB para asegurar el buen funcionamiento con 1000 usuarios concurrentes.

4.2.1.1 Alcances

En términos generales el sistema está dirigido a permitir la consulta en las etapas de formulación y expedición, apoyar la etapa de ejecución con la automatización de reportes y la de evaluación al proveer un mecanismo de registro de monitoreo y visualización de indicadores de desempeño.

4.2.1.2 Objetivos generales

Los objetivos generales del sistema son:

1) Proporcionar un mecanismo de conocimiento y soporte geográfico de decisiones para la gestión del POETY. (QUÉ)

2) Desarrollando un Sistema que permita la gestión del POETY mediante la organización dinámica de información relevante, la simplificación de informes técnicos, al definirse como base para configuración de la bitácora ambiental, así como geovisualizaciones que faciliten los procesos multi-escalares, multi-temporales y multi-sectoriales de la transformación territorial y la vulnerabilidad de los ecosistemas al cambio climático. (CÓMO)

3) Para que las autoridades, así como otros actores de la vida pública cuenten con un mecanismo que funcione como sistema de soporte geográfico de decisiones para la puesta en marcha de la actualización del POETY, para el manejo, análisis y visualización de información que facilite la gobernanza colaborativa en el proceso de ordenamiento ecológico en la entidad y su articulación con otros instrumentos de planeación pertinentes. (POR QUÉ)

4.2.1.3 Requerimientos

En la Tabla 4.2.1 se muestran las capacidades del sistema para cada usuario. El término “proyectos” se refiere a proyectos con autorizacion en materia de impacto ambiental (ejemplo: Manifestación de Impacto Ambiental). El sistema deberá conectarse a la base de datos del gobierno de Yucatán donde se aloja la información de los proyectos con autorizacion en materia de impacto ambiental, para extraer los datos necesarios que formarán parte de las fichas de UGAs, de los reportes de criterios de regulación por polígono, y de los reportes de monitoreo y desempeño del programa de ordenamiento.

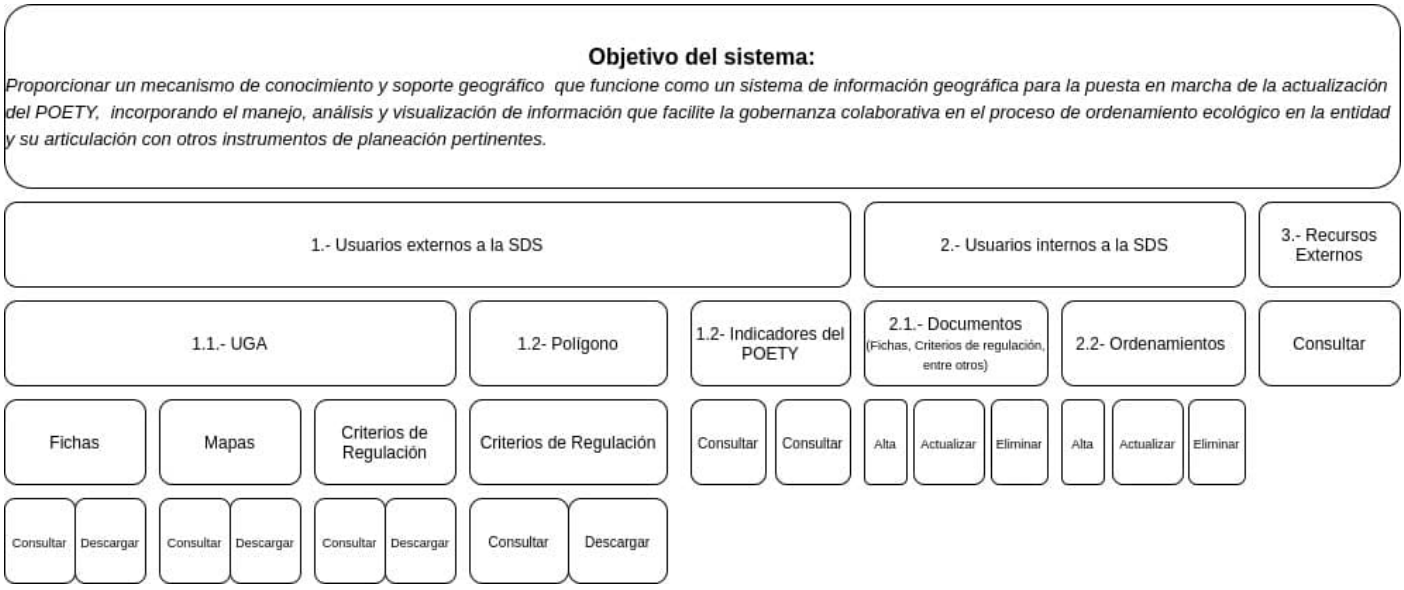


Figura 4.2.1 Capacidades del sistema por usuario

A continuación, se detallan los requerimientos y objetivos que el sistema cumplirá. En atención a los términos de referencia es preciso que el sistema cumpla con los requerimientos presentados en la **Tabla 4.2.1**

Tabla 4.2.1 Requerimientos del Sistema

Num	Requerimiento
1	Acceso interactivo a bancos de datos de la caracterización ambiental y socio-económica
2	Monitoreo y evaluación pública a través de índices e indicadores del desempeño del ordenamiento ecológico
3	Consulta interactiva de mecanismos de resolución de conflictos del programa de ordenamiento ecológico con base en los criterios, estrategias y lineamientos de regulación ecológico en cada UGA
4	Simplificar la consulta de informes técnicos
5	Permitir la realización de procedimientos de geovisualización que incluyan como mínimo manejo de bancos de datos, manejo de modelos de análisis, generador de reportes gráficos y tabulares e interfaces de operación

La Figura 4.2.1 descompone estos requerimientos en función a los actores del sistema y a las acciones que pueden realizar dentro del mismo. Esto permite la generación de subsistemas con el objetivo de modularizar el diseño del sistema.

4.2.1.4 Beneficios

El sistema permitirá: a) organizar de manera dinámica toda la información relevante del POETY; b) simplificar la consulta de informes técnicos; c) dar elementos de sustento en los procedimientos administrativos para la emisión de permisos, licencias y autorizaciones; d) facilitar la gobernanza colaborativa del ordenamiento ecológico; e) servir de base para la configuración de la bitácora ambiental del POETY; f) permitir la realización de procedimientos de geovisualización que incluyan manejo de bancos de datos, manejo de modelos de análisis, generador de reportes gráficos y tabulares e interfaces de operación; y g) facilitar la exploración y geovisualización de los procesos multi-escalares, multitemporales y multi-sectoriales de la transformación territorial y la vulnerabilidad de los ecosistemas al cambio climático.

Adicionalmente el sistema integrará la capacidad de agregar nuevos ordenamientos ecológicos (OE), por ejemplo, ordenamientos locales (POELs) o costero (POETCY), dando así la capacidad a la Secretaría de Desarrollo Sustentable del Gobierno del Estado de Yucatán (SDS) de agregar en un mismo instrumento de consulta y gestión los distintos OEs del territorio. Un beneficio directo de este esquema será que al consultar los criterios de regulación que aplican a un polígono particular, el sistema responderá con información de todos los OEs que apliquen a la zona, es decir si el polígono en cuestión intersecta con Unidades de gestión ambiental (UGAs) estatales y/o municipales y/o costeras, en una sola consulta, se obtendrán todos los criterios de regulación que apliquen en los distintos niveles. Esta capacidad se implementará de modo tal que además de servir como insumo de la bitácora de OEs se pueda utilizar como servicio independiente en otros flujos de trabajo de la SDS.

4.2.1.5 Limitaciones

Si bien el sistema incluye la capacidad de agregar nuevos ordenamientos ecológicos, como por ejemplo ordenamientos locales (POELs) o costero (POETCY), los datos y resultados que se incorporen al sistema tendrán que ajustarse a lineamientos y formatos específicos.

4.2.1.6 Casos de uso

En esta sección se describen los casos de uso, los módulos a los que pertenecen y los diagramas de casos de uso. Los diagramas de casos de uso proporcionan una descripción visual e integral de los requisitos de un sistema, como se puede observar en las Figura 4.2.2 y Figura 4.2.3 los diagramas incorporan actores, casos de uso y las relaciones entre estos.

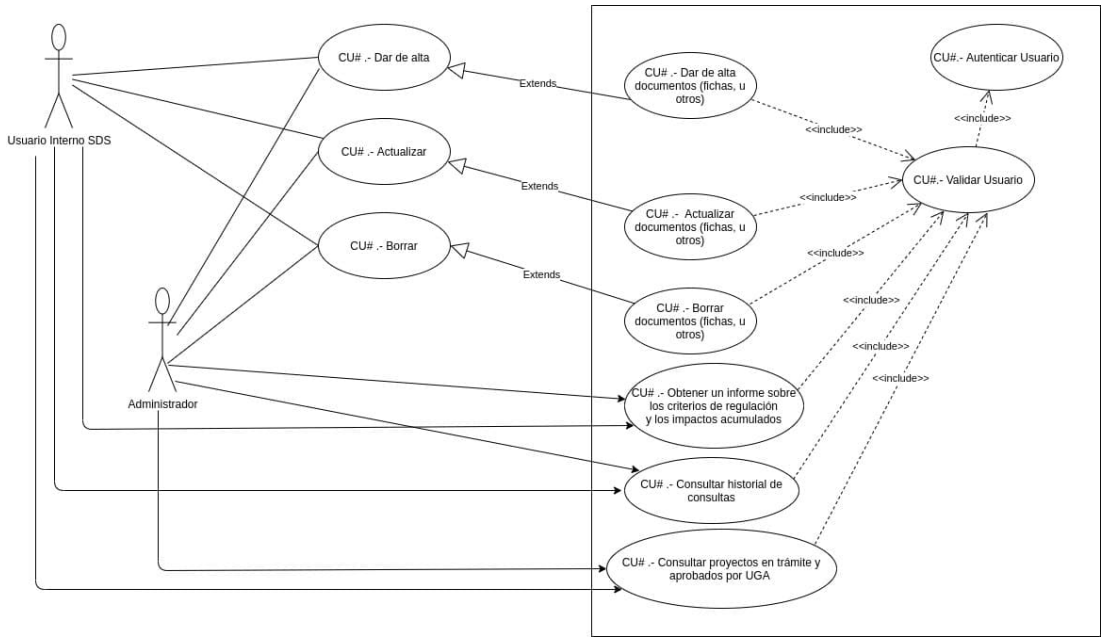


Figura 4.2.2 Diagrama de administración de recursos

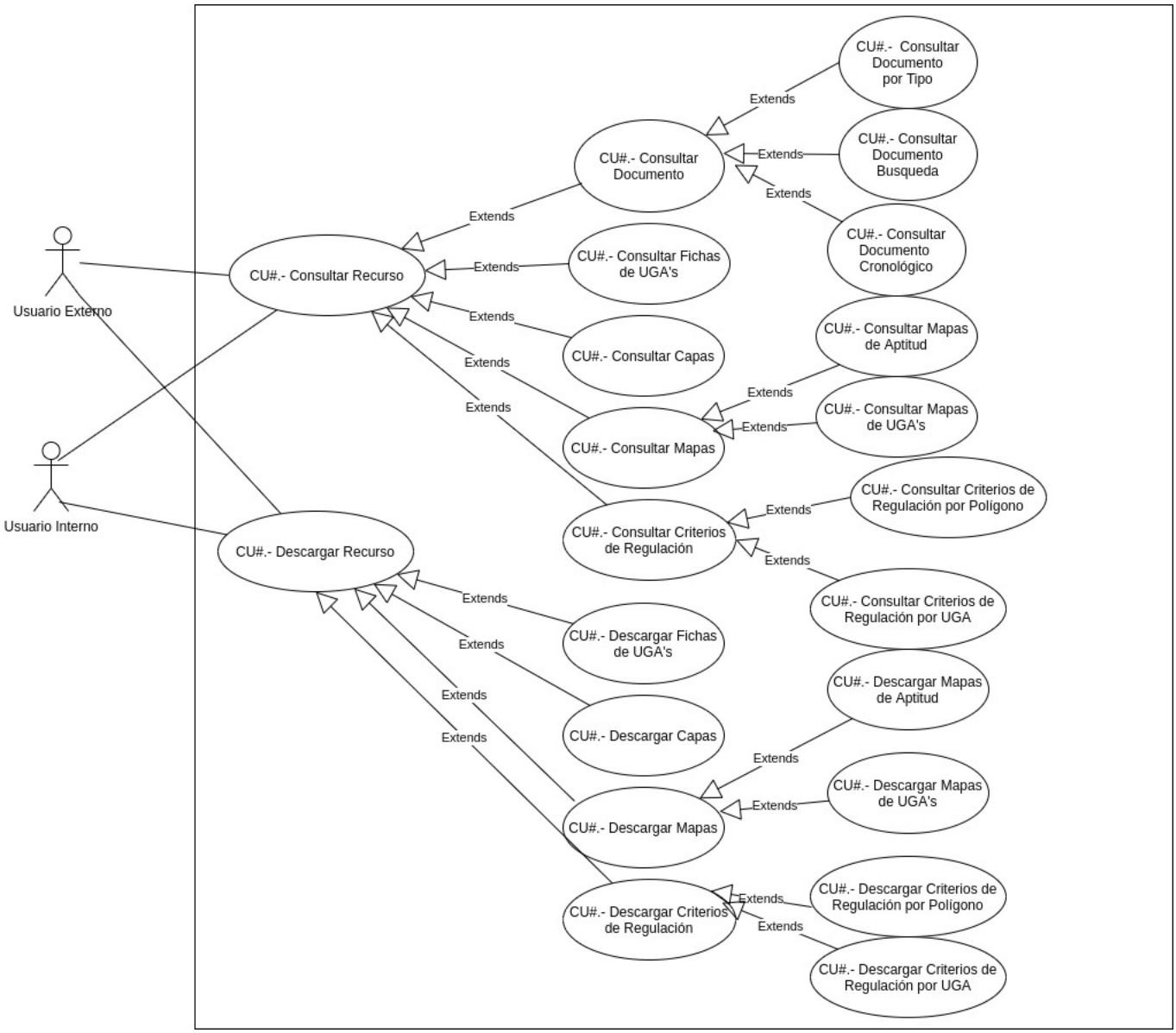


Figura 4.2.3 Diagrama de consumo de recursos

Para dividir el desarrollo de las diferentes partes del sistema, se separan los casos de uso en módulos por su funcionalidad (ver Tabla 4.2.2).

Tabla 4.2.2 Módulos y Casos de Uso

Módulo	Casos de Uso	Permiso UI[a]	Permiso UE[b]	Permiso A[c]
Consultas y descargas	Consultar fichas de UGAs	■	■	■
	Descargar ficha de UGA	■	■	■
	Consultar capas de insumo para el ordenamiento	■	■	■
	Consultar mapas de aptitud	■	■	■
	Descargar capas	■	■	■
	Consultar mapas de UGAs	■	■	■
	Consultar documentos relacionados con la formulación del (POETY, POETCY, o POEL)	■	■	■
	Consultar documentos del (POETY, POETCY, o POEL) por búsqueda	■	■	■
	Consultar documentos del (POETY, POETCY, o POEL) por orden cronológico	■	■	■
	Consultar documentos del (POETY, POETCY, o POEL) por tipo	■	■	■
	Descargar documento	■	■	■
	Consultar criterios de regulación por polígono	■	■	■
	Descargar criterios de regulación por polígono	■	■	■
	Descargar criterios de regulación por UGA	■	■	■
	Consultar indicadores del (POETY, POETCY, o POEL)	■	■	■
Actualización de recursos	Dar de alta documentos (fichas, u otros)	■		■
	Actualizar documentos (fichas, u otros)	■		■
	Borrar documentos (fichas, u otros)	■		■
	Consultar historial de reportes por polígono	■		■
	Consultar proyectos en trámite y aprobados por UGA	■		■
Automatización de reportes	Obtener un informe sobre los criterios de regulación y los impactos acumulados	■		■
Módulo de administración	Dar de alta usuario interno de SDS			■
	Dar de baja usuario interno de SDS			■
	Modificar datos de usuarios internos de SDS			■
	Validar usuraio			■
	Autenticar usuario			■
	Dar de alta un ordenamiento			■
	Borrar un ordenamiento			■
	Modificar datos generales de un ordenamiento			■

[a] Usuario Interno, [b] Usuario Externo, [c] Administrador

Un caso de uso documenta las interacciones de un actor con el sistema en el contexto de una acción específica. Los casos de uso describen lo que el sistema hace, pero no especifican como lo hace. A continuación, se describe a detalle los casos uso, esto incluye el flujo principal, los actores participantes, los flujos alternativos y los flujos de excepción.

Tabla 4.2.3 Caso de uso: Consultar fichas de UGAs

Elemento	Descripción
Caso de uso (CU)	Consultar fichas de UGAs
Id del CU	1
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario selecciona la UGA en el mapa, solicita la ficha de esa UGA y el sistema despliega la visualización de la ficha
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento **Del sistema** La ficha de la UGA está disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario seleciona la capa de UGAs 4. El usuario selecciona la UGA de interés 5. El usuario solicita la ficha de la UGA de interés 6. El usuario descarga la ficha(A1)
Flujos alternos	(A1) El usuario solicita la descarga de la ficha (CU1) (A2) El usuario solicita la descarga de los criterios de regulación por UGA (CU13)
Flujos de excepción	(E1) En caso de no existir la ficha de la UGA solicitada, levantar un reporte
Post-condiciones	

Tabla 4.2.4 Caso de uso: Descargar ficha de UGA

Elemento	Descripción
Caso de uso (CU)	Descargar ficha de UGA
Id del CU	2
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario descarga la ficha técnica de una unidad de gestión ambiental
Pre-condiciones	**Del proceso**: El usuario realiza la consulta de una UGA para visualizar su ficha **Del sistema** La ficha de la UGA está disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario la capa de UGAs 4. El usuario solicita la ficha de la UGA de interés 5. El usuario descarga la ficha
Flujos alternos	
Flujos de excepción	(E1) En caso de no existir la ficha de la UGA solicitada, levantar un reporte
Post-condiciones	

Tabla 4.2.5 Caso de uso: Consultar capas de insumo para el ordenamiento

Elemento	Descripción
Caso de uso (CU)	Consultar capas de insumo para el ordenamiento
Id del CU	3
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario consulta las capas de insumo que conforman el mapa de aptitud de un sector específico
Pre-condiciones	**Del proceso**: El usuario selecciona un sector **Del sistema** Las capas de los insumos están disponibles y tienen un estilo asociado
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selcciona la sección de diagnóstico 4. El sistema muestra una lista de sectores 5. El usuario elige el sector de interés 6. El sistema visualiza el mapa de aptitud del sector (CU3) 7. El usuario selecciona la capa de insumo deseada 8. El sistema muestra la capa deseada y la función de valor asociada
Flujos alternos	(A1) El usuario descarga la capa de insumo (CU4)
Flujos de excepción	(E1) En caso de no existir la capa de insumo solicitada, levantar un reporte
Post-condiciones	

Tabla 4.2.6 Caso de uso: Consultar mapas de aptitud

Elemento	Descripción
Caso de uso (CU)	Consultar mapas de aptitud
Id del CU	4
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario visualiza el mapa de aptitud por sector
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento **Del sistema** La capa está disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selcciona la sección de diagnóstico 4. El sistema muestra una lista de sectores 5. El usuario elige el sector de interés 6. El sistema visualiza el mapa de aptitud del sector y despliega una lista de los insumos
Flujos alternos	(A1) El usuario visualiza las capas de insumo (CU2)
Flujos de excepción	(E1) En caso de no visualizar el mapa, levantar un reporte
Post-condiciones	

Tabla 4.2.7 Caso de uso: Descargar capas

Elemento	Descripción
Caso de uso (CU)	Descargar capas
Id del CU	5
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario descarga la capa visualizada
Pre-condiciones	**Del proceso**: La capa esta visualizada en el sistema **Del sistema** La capa está disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selcciona la sección de diagnóstico 4. El usuario visualiza una capa mediante los casos de uso (CU2,CU3,CU5) 5. El usuario solicita la descarga 6. El sistema incluye en la descarga la capa geográfica y los metadatos asociados
Flujos alternos	NA
Flujos de excepción	(E1) En caso de no descargarse la capa solicitada, levantar un reporte
Post-condiciones	

Tabla 4.2.8 Caso de uso: Consultar mapas de UGAs

Elemento	Descripción
Caso de uso (CU)	Consultar mapas de UGAs
Id del CU	6
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario visualiza las unidades de gestión ambiental
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento **Del sistema** La ficha de la UGA está disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario la capa de UGAs 4. El sistema muestra la capa de UGAs
Flujos alternos	(A1) El usuario solicita la descarga de la capa de las ugas (CU4)
Flujos de excepción	(E1) En caso de no visualizar el mapa solicitado, levantar un reporte
Post-condiciones	

Tabla 4.2.9 Caso de uso: Consultar documento relacionado con la formulación del (POETY, POETCY, o POEL)

Elemento	Descripción
Caso de uso (CU)	Consultar documento relacionado con la formulación del (POETY, POETCY, o POEL)
Id del CU	7
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario consulta cualquier documento relacionado con la formulación del POETY (por búsqueda, por orden cronológico, por categoría)
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento. **Del sistema** El documento este disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selecciona la sección de documentos de la formulación del ordenamiento específico 4. El sistema despliega los modos de búsqueda (por búsqueda de palabra clave,por orden cronológico, por tipo) Ver casos de uso (CU7, CU8, CU9)
Flujos alternos	NA
Flujos de excepción	NA
Post-condiciones	

Tabla 4.2.10 Caso de uso: Consultar documento del (POETY, POETCY, o POEL) por búsqueda

Elemento	Descripción
Caso de uso (CU)	Consultar documento del (POETY, POETCY, o POEL) por búsqueda
Id del CU	8
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario consulta un documento relacionado con la formulación del POETY por búsqueda de palabra(s) clave(s)
Pre-condiciones	**Del proceso**: El usuario elige el modo de búsqueda por palabra clave (CU6) **Del sistema** El documento este disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selecciona la sección de documentos de la formulación del ordenamiento específico 4. El usuario selecciona el modo de búsqueda de palabra clave 5. El usuario ingresa la palabra(s) clave relacionada(s) con el documento 6. El sistema despliega una lista de documentos relacionados 7. El usuario selecciona el documento deseado 8. El sistema ejecuta la visualización del documento seleccionado
Flujos alternos	(A1) El usuario solicita la descarga del documento (CU10)
Flujos de excepción	NA
Post-condiciones	

Tabla 4.2.11 Caso de uso: Consultar documento del (POETY, POETCY, o POEL) por orden cronológico

Elemento	Descripción
Caso de uso (CU)	Consultar documento del (POETY, POETCY, o POEL) por orden cronológico
Id del CU	9
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario consulta un documento relacionado con la formulación del POETY por orden cronológico
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento. **Del sistema** El documento este disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selecciona la sección de documentos de la formulación del ordenamiento específico 4. El usuario selecciona el modo de búsqueda por orden cronológico 5. El sistema despliega una lista de documentos por orden cronológico 6. El usuario selecciona el documento deseado 7. El sistema ejecuta la visualización del documento seleccionado
Flujos alternos	(A1) El usuario solicita la descarga del documento (CU10)
Flujos de excepción	NA
Post-condiciones	

Tabla 4.2.12 Caso de uso: Consultar documento del (POETY, POETCY, o POEL) por tipo

Elemento	Descripción
Caso de uso (CU)	Consultar documento del (POETY, POETCY, o POEL) por tipo
Id del CU	10
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario consulta un documento relacionado con la formulación del POETY por tipo
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento. **Del sistema** El documento este disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selecciona la sección de documentos de la formulación del ordenamiento específico 4. El usuario selecciona el modo de búsqueda por tipo 5. El sistema despliega una lista de documentos por tipo de documento 6. El usuario selecciona el tipo de docuemento 7. El sistema despliega una lista de los documentos en la categoria seleccionada 8. El usuario selecciona el documento deseado 9. El sistema ejecuta la visualización del documento deseado
Flujos alternos	(A1) El usuario solicita la descarga del documento (CU10)
Flujos de excepción	NA
Post-condiciones	

Tabla 4.2.13 Caso de uso: Descargar documento

Elemento	Descripción
Caso de uso (CU)	Descargar documento
Id del CU	11
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	EL usuario descarga el documento relacionado con la formulación del POETY
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento. **Del sistema** El documento este disponible
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario selecciona la sección de documentos de la formulación del ordenamiento específico 4. El usuario selecciona el modo de búsqueda 5. El sistema despliega una lista de documentos según el tipo de búsqueda (CU6) 6. El sistema despliega una lista de los documentos 7. El usuario selecciona el documento deseado 8. El sistema ejecuta la visualización del documento deseado 9. El usuario solicita la descarga del documento
Flujos alternos	NA
Flujos de excepción	NA
Post-condiciones	

Tabla 4.2.14 Caso de uso: Consultar criterios de regulación por polígono

Elemento	Descripción
Caso de uso (CU)	Consultar criterios de regulación por polígono
Id del CU	12
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario consulta los criterios de regulación que aplican en un área de interés
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento **Del sistema** Ninguna
Flujo principal	1. El usuario accede a la bitácora del ordenamiento 2. El usuario selecciona la sección de criterios de regulación 3. El usuario provee un poligono de interés en formato shapefile 4. El sistema despliega un mapa con la intersección del polígono proporcionado y las UGAS estatales y/o locales y una lista de los criterios de regulación que aplican para cada una de las interesecciones
Flujos alternos	(A1) El usuario solicita la descarga de los criterios de regulación (CU12)
Flujos de excepción	(E1) En caso de que el shapefile proporcionado sea defectuoso el sistema muestra un mensaje çapa no válidaz regresa a la sección de criterios de regulación (E2) En caso de no aplicar ningún criterio de regulación, el sistema responde que no aplica ningún criterio y regresa a la sección de criterios de regulación
Post-condiciones	

Tabla 4.2.15 Caso de uso: Descargar criterios de regulación por polígono

Elemento	Descripción
Caso de uso (CU)	Descargar criterios de regulación por polígono
Id del CU	13
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario descarga los criterios de regulación que aplican en un polígono específico
Pre-condiciones	**Del proceso**: El usuario realiza la consulta de criterios de regulación por polígono (CU11) **Del sistema** Ninguna
Flujo principal	1. El usuario accede a la bitácora del ordenamiento 2. El usuario selecciona la sección de criterios de regulación 3. El usuario provee un poligono de interés en formato shapefile 4. El sistema despliega un mapa con la intersección del polígono proporcionado y las UGAS estatales y/o locales y una lista de los criterios de regulación que aplican para cada una de las interesecciones 5. El usuario descarga un documento que incluye el mapa y la lista de los criterios de regulación o solo la lista de criterios de regulación en formato csv o xls
Flujos alternos	NA
Flujos de excepción	(E1) En caso de error al descargar la lista solicitada, levantar un reporte
Post-condiciones	

Tabla 4.2.16 Caso de uso: Descargar criterios de regulación por UGA

Elemento	Descripción
Caso de uso (CU)	Descargar criterios de regulación por UGA
Id del CU	14
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario descarga los criterios de regulación que aplican en una unidad de gestión ambiental
Pre-condiciones	**Del proceso**: El usuario visualiza la ficha de la UGA **Del sistema** Los criterios de la UGA están disponibles
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario la capa de UGAs 4. El usuario selecciona la UGA de interes 5. El usuario solicita la ficha de la UGA de interes 6. El usuario descarga los criterios de regulación en formato csv o xls
Flujos alternos	NA
Flujos de excepción	(E1) En caso de error al descargar la lista solicitada, levantar un reporte
Post-condiciones	

Tabla 4.2.17 Caso de uso: Consultar indicadores del (POETY, POETCY, o POEL)

Elemento	Descripción
Caso de uso (CU)	Consultar indicadores del (POETY, POETCY, o POEL)
Id del CU	15
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario consulta indicadores del POETY.
Pre-condiciones	**Del proceso**: Debe existir el indicador a consultar. **Del sistema** Cualquier usuario puede acceder al recurso.
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario se dirige al apartado de "Indicadores del (POETY, POETCY, o POEL)". 4. El usuario ingresa el id del indicador a consultar. (E1) 5. El usuario descarga la ficha del indicador consultado. 6. El Sistema muestra una mensaje de confirmación sobre el indicador. 7. El usuario acepta la advertencia. (A1) 8. La información se descarga en un archivo PDF o CSV.
Flujos alternos	(A1) En caso contrario, el usuario sale del proceso.
Flujos de excepción	(E1). El sistema no puede acceder a la base de datos.
Post-condiciones	

Tabla 4.2.18 Caso de uso: Dar de alta documentos (fichas, u otros)

Elemento	Descripción
Caso de uso (CU)	Dar de alta documentos (fichas, u otros)
Id del CU	16
Actores participantes	- Usuario interno
Breve descripción	El usuario sube un documento relacionado con la formulación del POETY o una ficha de UGA
Pre-condiciones	**Del proceso**: El usuario se autentifica y tiene los permisos necesarios **Del sistema** Ninguna
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario se autentifica con nombre de usuario y contraseña 3. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 4. El usuario selecciona la sección a la que pertenece el documento 5. El usuario proporciona el titulo del documento 6. El usuario proporciona una descripción general del documento 7. El usuario adjunta el documento
Flujos alternos	NA
Flujos de excepción	(E1) En caso de error en la autentificación el sistema redirige a la recuperación de contraseña
Post-condiciones	

Tabla 4.2.19 Caso de uso: Actualizar documentos (fichas, u otros)

Elemento	Descripción
Caso de uso (CU)	Actualizar documentos (fichas, u otros)
Id del CU	17
Actores participantes	- Usuario interno
Breve descripción	El usuario actualiza un documento relacionado con la formulación del POETY o una ficha de UGA
Pre-condiciones	**Del proceso**: El usuario se autentifica y tiene los permisos necesarios **Del sistema** Ninguna
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario se autentifica con nombre de usuario y contraseña 3. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 4. El usuario selecciona la sección a la que pertenece el documento 5. El sistema despliega una lista de los documentos 6. El usuario selecciona el documento a actualizar 7. El usuario actualiza el titulo, la descripción o el documento
Flujos alternos	NA
Flujos de excepción	(E1) En caso de error en la autentificación el sistema redirige a la recuperación de contraseña
Post-condiciones	

Tabla 4.2.20 Caso de uso: Borrar documentos (fichas, u otros)

Elemento	Descripción
Caso de uso (CU)	Borrar documentos (fichas, u otros)
Id del CU	18
Actores participantes	- Usuario interno
Breve descripción	El usuario borra un documento relacionado con la formulación del POETY o una ficha de UGA
Pre-condiciones	**Del proceso**: El usuario se autentifica y tiene los permisos necesarios **Del sistema** La capa que se quiere actualizar existe
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario se autentifica con nombre de usuario y contraseña 3. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 4. El usuario selecciona la sección a la que pertenece el documento 5. El sistema despliega una lista de los documentos 6. El usuario selecciona el documento a borrar 7. El usuario borra el documento 8. El sistema manda un mensaje para confirmar la acción de borrar 9. El documento ha sido borrado
Flujos alternos	NA
Flujos de excepción	(E1) En caso de error en la autentificación el sistema redirige a la recuperación de contraseña
Post-condiciones	

Tabla 4.2.21 Caso de uso: Consultar historial de reportes por polígono

Elemento	Descripción
Caso de uso (CU)	Consultar historial de reportes por polígono
Id del CU	19
Actores participantes	- Administrador - Usuario Interno de la SDS
Breve descripción	El usuario consulta el historial de consultas
Pre-condiciones	**Del proceso**: Los usuarios deben estar registrados en el sistema **Del sistema** Ya se han realizado consultas al sistema y el usuario tiene permisos apropiados para visualizarlas
Flujo principal	1. El usuario se autentifica (CU33). 2. El usuario selecciona el ordenamiento deseado (POETY,POETCY, o POEL) 3. El usuario se dirige al apartado de "Consultar historial de consultas". 4. El sistema despliega una lista de consultas por orden Cronológico. 5. El usuario selecciona una consulta. 6. El sistema genera el archivo PDF de dicha consulta. (A2)
Flujos alternos	(A1) Las credenciales proporcionadas son incorrectas. Se informa al usuario y se le solicitan nuevamente sus credenciales. (A2) El usuario solicita la descarga del PDF
Flujos de excepción	
Post-condiciones	

Tabla 4.2.22 Caso de uso: Consultar proyectos en trámite y aprobados por UGA

Elemento	Descripción
Caso de uso (CU)	Consultar proyectos en trámite y aprobados por UGA
Id del CU	20
Actores participantes	- Administrador - Usuario Interno de la SDS
Breve descripción	El usuario consulta y visualiza los proyectos en trámite y aprobados por UGA
Pre-condiciones	**Del proceso**: Los usuarios deben estar registrados en el sistema **Del sistema** Ya cuentan en la base de datos del sistema registros de proyectos. Los usuarios tiene permisos apropiados para consultar y visualizar los proyectos.
Flujo principal	1. El usuario se autentifica (CU33). 2. El usuario selecciona el ordenamiento deseado (POETY,POETCY, o POEL) 3. El usuario se dirige al apartado de "Consultar proyectos en trámite y aprobados". 4. El sistema despliega el listado de UGAs . 5. El usuario selecciona una UGA. 6. El sistema visualiza el polígono de la UGA así como también los proyectos (como puntos) donde se le asigna un color de acuerdo si están aprobados o en trámite.
Flujos alternos	(A1) Las credenciales proporcionadas son incorrectas. Se informa al usuario y se le solicitan nuevamente sus credenciales.
Flujos de excepción	
Post-condiciones	

Tabla 4.2.23 Caso de uso: Obtener un informe sobre los criterios de regulación y los impactos acumulados

Elemento	Descripción
Caso de uso (CU)	Obtener un informe sobre los criterios de regulación y los impactos acumulados
Id del CU	21
Actores participantes	- Usuario externo - Usuario interno
Breve descripción	El usuario obtiene informes sobre los criterios de regulación y los impactos acumulados.
Pre-condiciones	**Del proceso**: Debe existir el informe sobre los criterios de regulación y los impactos acumulados a consultar. **Del sistema** Cualquier usuario puede acceder al recurso.
Flujo principal	1. El usuario accede a la bitácora del ordenamientos 2. El usuario selcciona el ordenamiento deseado (POETY, POETCY, o POEL) 3. El usuario se dirige al apartado de "Informes sobre los criterios de regulación y los impactos acumulados". 4. El usuario ingresa el id del informe a consultar. (E1) 5. El usuario descarga la ficha del informe consultado. 6. El Sistema muestra una mensaje de confirmación sobre el informe. 7. El usuario acepta la advertencia. (A1) 8. La información se descarga en un archivo PDF o CSV.
Flujos alternos	(A1) En caso contrario, el usuario sale del proceso.
Flujos de excepción	(E1) El sistema no puede acceder a la base de datos.
Post-condiciones	

Tabla 4.2.24 Caso de uso: Dar de alta Usuario interno

Elemento	Descripción
Caso de uso (CU)	Dar de alta Usuario interno
Id del CU	22
Actores participantes	- Usuario de interno - Sistema interno
Breve descripción	El caso de uso describe el proceso de registro de un nuevo usuario de la SDS en el sistema. Los datos del usuario deben ser verificados en el sistema de la SDS para que el usuario sea registrado.
Pre-condiciones	**Del proceso**: El usuario interno de la SDS aún no ha sido registrado en el sistema. **Del sistema** El usuario interno debe contar con un número de trabajador y estar en activo.
Flujo principal	1. El usuario solicita interno al sistema registrarse como usuario. 2. El sistema le solicita su número de trabajador. 3. El usuario proporciona su número de trabajador. 4. El sistema verifica la información del trabajador dentro del sistema interno de la SDS.(A1)(E1) 5. El sistema verifica la información del trabajador dentro del sistema.(A2) 6. El sistema solicita un nombre de usuario y contraseña. 7. El sistema guarda la información proporcionada.(A3) 8. El sistema confirma el registro del usuario interno.
Flujos alternos	(A1) Si el trabajador no se encuentra activo, el sistema se lo informa y el proceso termina. (A2) Si un usuario ya ha sido dado de alta con ese número de trabajador, el sistema se lo informa y el proceso termina. (A3) Si el nombre de usuario coincide con uno que ya ha sido registrado, el sistema se lo informa al usuario, indicando que tiene que cambiarlo.
Flujos de excepción	(E1). No se tiene acceso a la base de datos del sistema interno de la SDS.
Post-condiciones	

Tabla 4.2.25 Caso de uso: Dar de baja Usuario interno

Elemento	Descripción
Caso de uso (CU)	Dar de baja Usuario interno
Id del CU	23
Actores participantes	- Administrador
Breve descripción	El caso de uso describe el proceso de baja del sistema de un usuario interno de la SDS.
Pre-condiciones	**Del proceso**: El usuario a dar de baja debe estar registrado en el sistema. **Del sistema** El actor debe estar autenticado con el rol de Administrador
Flujo principal	1. El Administrador solicita al sistema el listado de los usuarios. 2. El sistema actualiza el listado de usuarios y lo muestra al Administrador.(E1). 3. El Administrador selecciona el usuario a dar de baja. 4. El Administrador indica al Sistema dar de baja a dicho usuario. 5. El Sistema solicita al usuario que indique si está seguro de la acción de dar de baja. 6. El Administrador confirma la solicitud (A1). 7. El Sistema da de baja al usuario seleccionado .(E2)
Flujos alternos	(A1). En caso contrario, el sistema regresa al Administrador a su interfaz de Administrador.
Flujos de excepción	(E1). El sistema no uede actualizar el listado de usuarios. El sistema muestra un mensaje al Administrador indicando que no es posible realizar la acción solicitada, pidiendo que lo intente más tarde. (E2) El sistema no puede eliminar o inhabilitar al usuario seleccionado. El sistema muestra un mensaje al Administrador indicando que no es posible realizar la acción solicitada, pidiendo que lo intente más tarde.
Post-condiciones	

Tabla 4.2.26 Caso de uso: Modificar datos de usuarios internos de SDS

Elemento	Descripción
Caso de uso (CU)	Modificar datos de usuarios internos de SDS
Id del CU	24
Actores participantes	- Usuario Interno
Breve descripción	El caso de uso describe el proceso en el que el usuario modifica sus datos y se guardan los cambios.
Pre-condiciones	**Del proceso**: El usuario a modificar debe estar registrado. **Del sistema** El usuario debe estar autenticado como usuario interno
Flujo principal	1. El usuario ingresa sus credenciales de usuario: nombre de usuario y contraseña.(A1) 2. El sistema presenta la interfaz de usuario. 3. El usuario se dirige al apartado de "Modificar datos de usuario". 4. El sistema muestra un listado con los datos actuales del usuario. 5. El usuario selecciona el dato particular a actualizar. 6. El Sistema muestra un formulario de actualización del dato 7. El usuario llena el formulario. 8. El Sistema muestra una advertencia de actualización. 9. El usuario acepta la advertencia.(A2) 10. El sistema registra los datos actualizados.(E1)
Flujos alternos	(A1) En caso de no contar con credenciales, ver caso de uso Dar de alta usuario. (A2) El usuario solicita al sistema que se cancelen los cambios hechos. El sistema descarta los cambios a los detalles del usuario seleccionado
Flujos de excepción	(E1) El sistema no puede actualizar los datos del usuario. El sistema muestra un mensaje al Usuario indicando que no es posible realizar la acción solicitada y pidiendo que lo intente más tarde
Post-condiciones	

Tabla 4.2.27 Caso de uso: Validar usuario

Elemento	Descripción
Caso de uso (CU)	Validar usuario
Id del CU	25
Actores participantes	- Administrador - Usuario interno de la SDS
Breve descripción	El usuario realiza una solicitud y se valida su identidad
Pre-condiciones	**Del proceso**: Ninguna, puede ocurrir en cualquier momento del proceso en que el actor requiera acceso autorizado. **Del sistema** El actor solicita acceso a un recurso protegido
Flujo principal	1. El sistema valida la autenticidad del usuario. (A1) 2. El sistema valida que los permisos del rol del usuario coincidan con los permisos requeridos para acceder al recurso solicitado. (A2)
Flujos alternos	(A1) El usuario no se encuentra autenticado: A) Se le solicita al usuario que se autentique para poder acceder. B) Se redirige al usuario a la pantalla de autenticación (Ver caso de uso: Autenticar usuario) (A2) El usuario autenticado no cuenta con los permisos requeridos para acceder al recurso solicitado: A) Se le indica al usuario que no tiene permisos suficientes. B) Se redirige al usuario a la pantalla de autenticación (Ver caso de uso: Autenticar usuario)
Flujos de excepción	
Post-condiciones	

Tabla 4.2.28 Caso de uso: Autenticar usuario

Elemento	Descripción
Caso de uso (CU)	Autenticar usuario
Id del CU	26
Actores participantes	- Administrador - Usuario Interno de la SDS
Breve descripción	El usuario ingresa sus credenciales para autenticar su identidad
Pre-condiciones	**Del proceso**: Los usuarios deben estar registrados en el sistema **Del sistema** El actor falla la validación al solicitar acceso a un recurso protegido.
Flujo principal	1. El sistema solicita las credenciales del usuario: nombre de usuario y contraseña. 2. El usuario ingresa su nombre de usuario y contraseña. 3. El sistema autentica al usuario por medio de sus credenciales (A1)
Flujos alternos	(A1) Las credenciales proporcionadas son incorrectas. Se informa al usuario y se le solicitan nuevamente sus credenciales.
Flujos de excepción	
Post-condiciones	

Tabla 4.2.29 Caso de uso: Dar de alta un ordenamiento

Elemento	Descripción
Caso de uso (CU)	Dar de alta un ordenamiento
Id del CU	27
Actores participantes	- Administrador - Usuario Interno de la SDS
Breve descripción	El usuario interno o administrador dan de alta un nuevo ordenamiento (POETY, POETCY, o POEL)
Pre-condiciones	**Del proceso**: Los usuarios deben estar registrados en el sistema **Del sistema** NA
Flujo principal	1. El usuario se autentifica (CU33). 2. El usuario entra al menu ordenamientos 3. El usuario selecciona la opción dar de alta 4. El ususario ingresa los datos generales del ordenamiento 5. El usuario guarda los cambios.
Flujos alternos	(A1) Las credenciales proporcionadas son incorrectas. Se informa al usuario y se le solicitan nuevamente sus credenciales.
Flujos de excepción	
Post-condiciones	

Tabla 4.2.30 Caso de uso: Borrar un ordenamiento

Elemento	Descripción
Caso de uso (CU)	Borrar un ordenamiento
Id del CU	28
Actores participantes	- Administrador - Usuario Interno de la SDS
Breve descripción	El usuario interno o administrador borran un ordenamiento existente (POETY, POETCY, o POEL)
Pre-condiciones	**Del proceso**: Los usuarios deben estar registrados en el sistema **Del sistema** NA
Flujo principal	1. El usuario se autentifica (CU33). 2. El usuario entra al menu ordenamientos 3. El usuario selecciona la opción borrar 4. El ususario selecciona el ordenamiento 5. El usuario guarda los cambios.
Flujos alternos	(A1) Las credenciales proporcionadas son incorrectas. Se informa al usuario y se le solicitan nuevamente sus credenciales.
Flujos de excepción	
Post-condiciones	

Tabla 4.2.31 Caso de uso: Modificar datos generales de un ordenamiento

Elemento	Descripción
Caso de uso (CU)	Modificar datos generales de un ordenamiento
Id del CU	29
Actores participantes	- Administrador - Usuario Interno de la SDS
Breve descripción	El usuario modifica los datos generales de un ordenamiento
Pre-condiciones	**Del proceso**: Los usuarios deben estar registrados en el sistema **Del sistema** El ordenamiento existe y el usuario tiene permisos apropiados para modificarlo
Flujo principal	1. El usuario se autentifica (CU33). 2. El usuario entra al menu ordenamientos 3. El usuario selecciona la opción editar 4. El ususario selecciona el ordenamiento 5. El usuario modifica alguno de los datos generales del ordenamiento 6. El usuario guarda los cambios.
Flujos alternos	(A1) Las credenciales proporcionadas son incorrectas. Se informa al usuario y se le solicitan nuevamente sus credenciales.
Flujos de excepción	
Post-condiciones	

4.2.1.7 Diagramas de actividad

Los diagramas de actividad están destinados a mostrar cualquier flujo o proceso en el sistema. Estos diagramas permiten modelar el flujo de información, creando un mapa visual de estos flujos, así como de las dependencias entre los casos de uso.

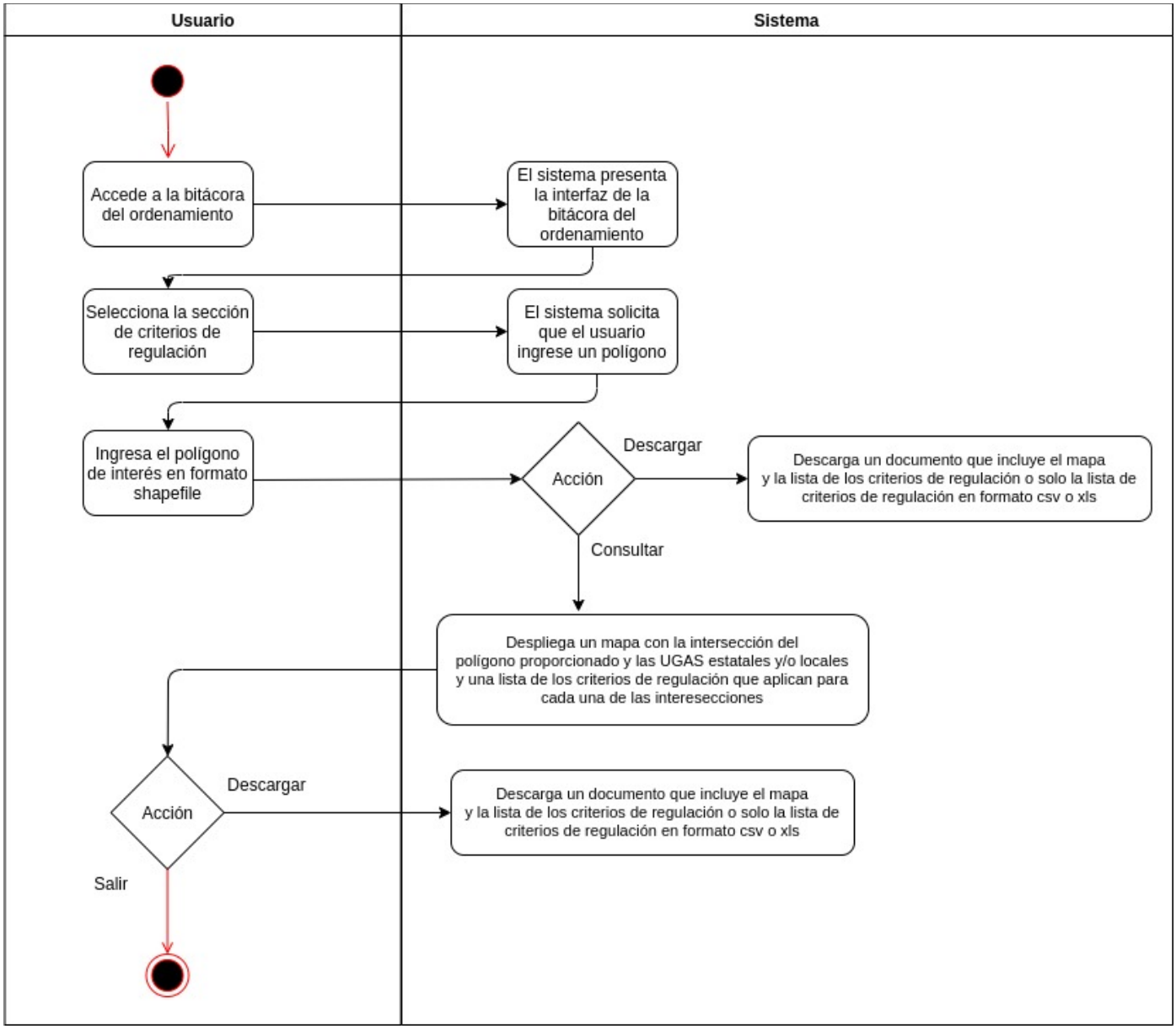


Figura 4.2.4 Diagrama de actividad consultar-descargar criterios de regulación por polígono

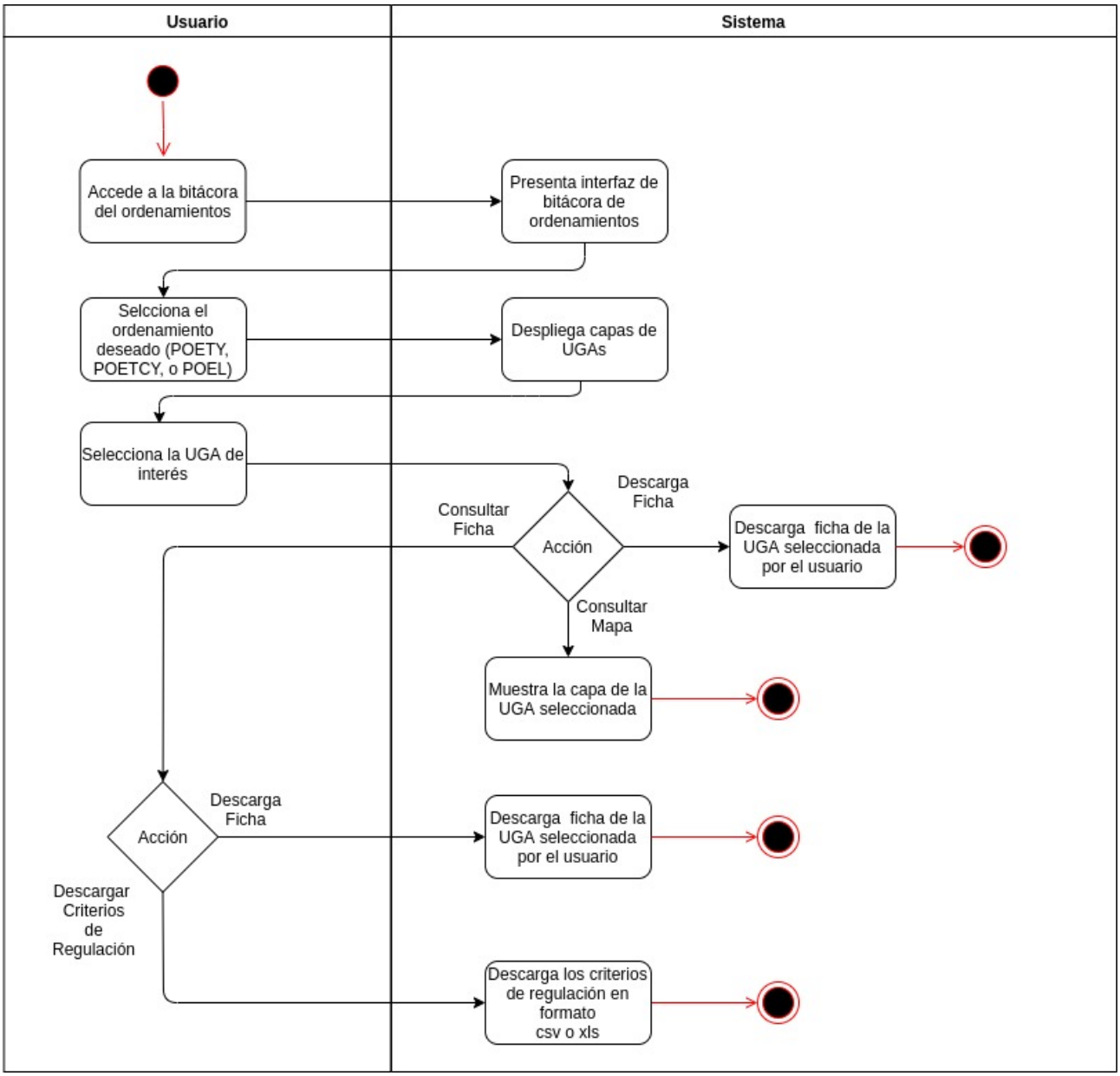


Figura 4.2.5 Diagrama de actividad consultar-descargar recursos por UGAs

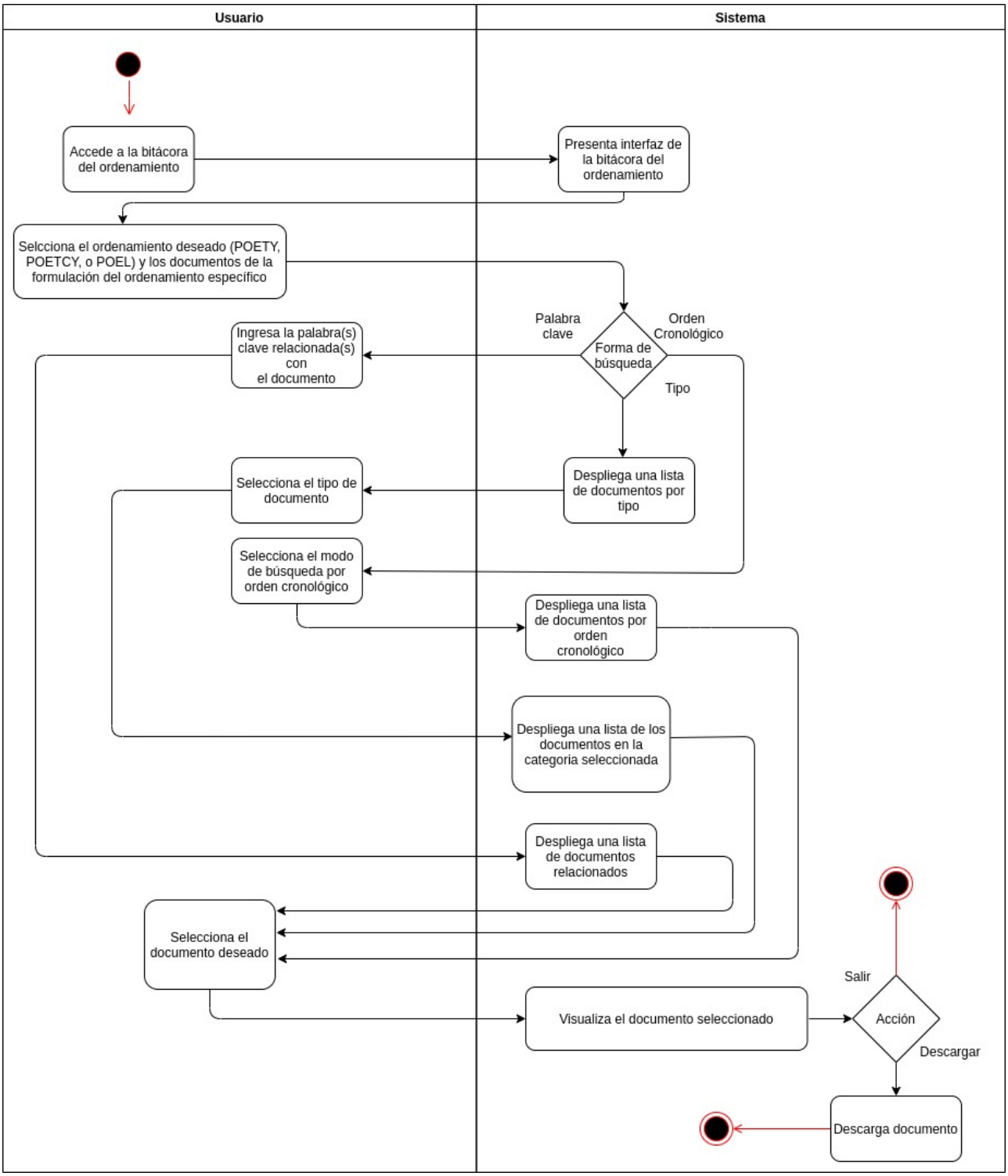


Figura 4.2.6 Diagrama de actividad consultar-descargar documentos relacionados con la formulación del POETY.

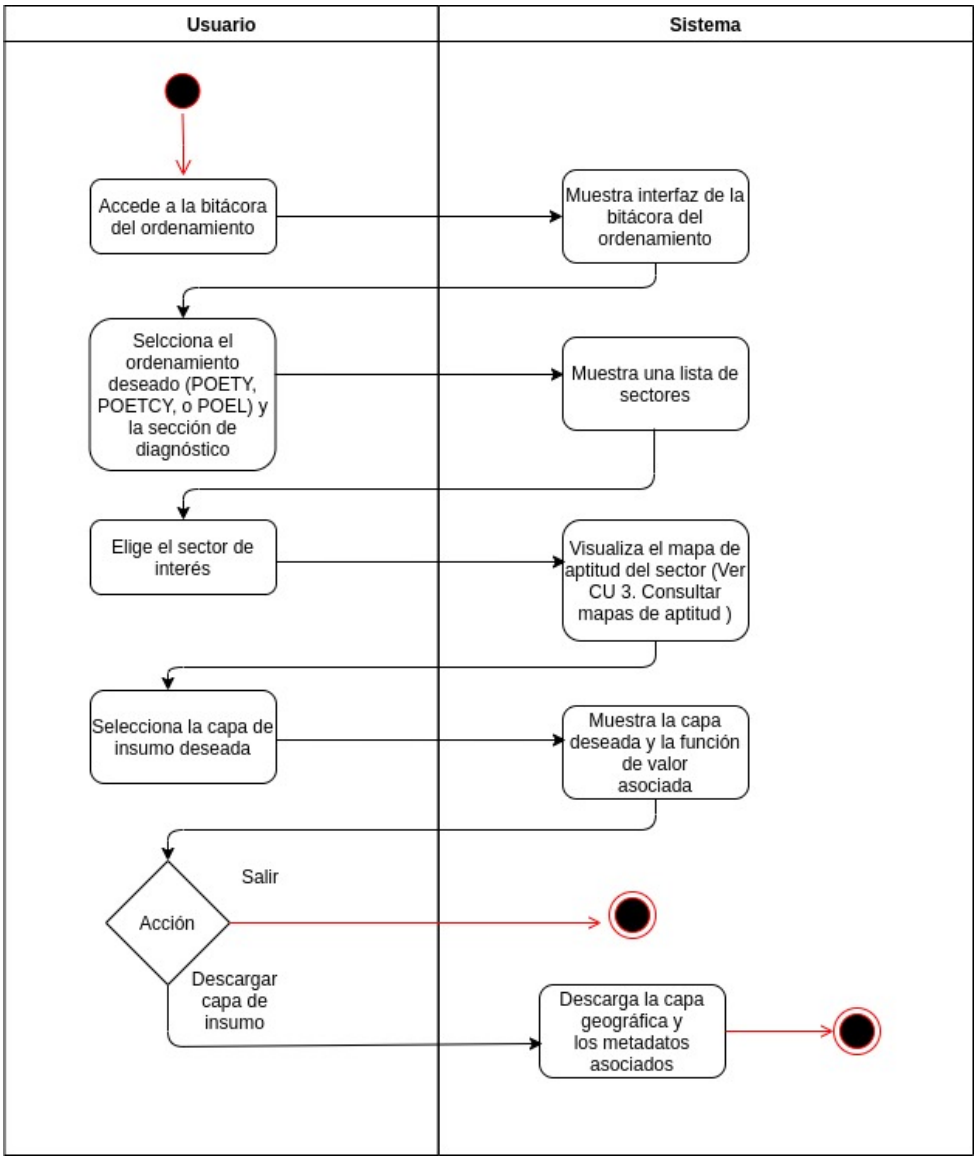


Figura 4.2.7 Diagrama de actividad consultar-descargar capas de insumo para el OE

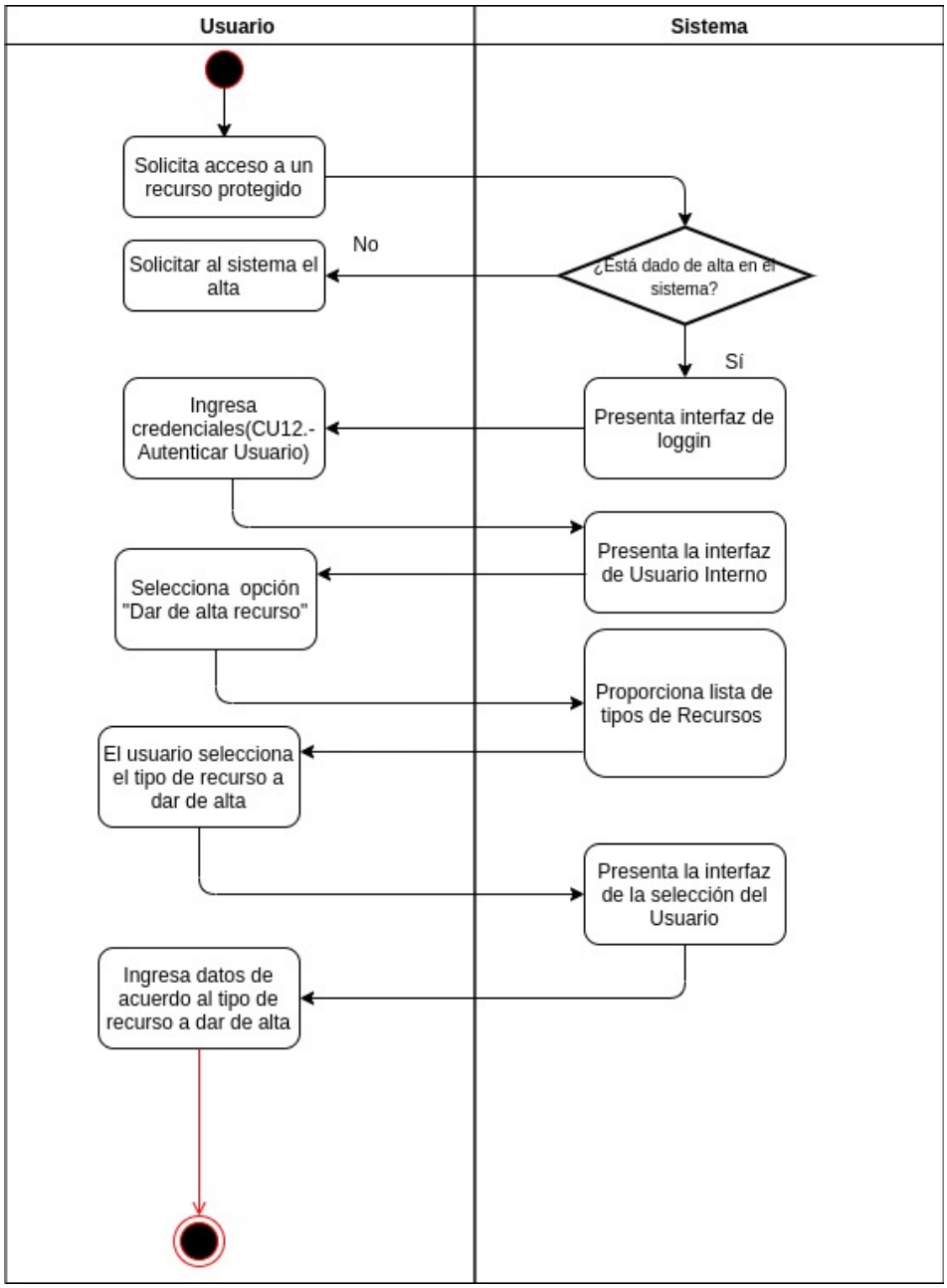


Figura 4.2.8 Diagrama de actividad Dar de Alta recurso

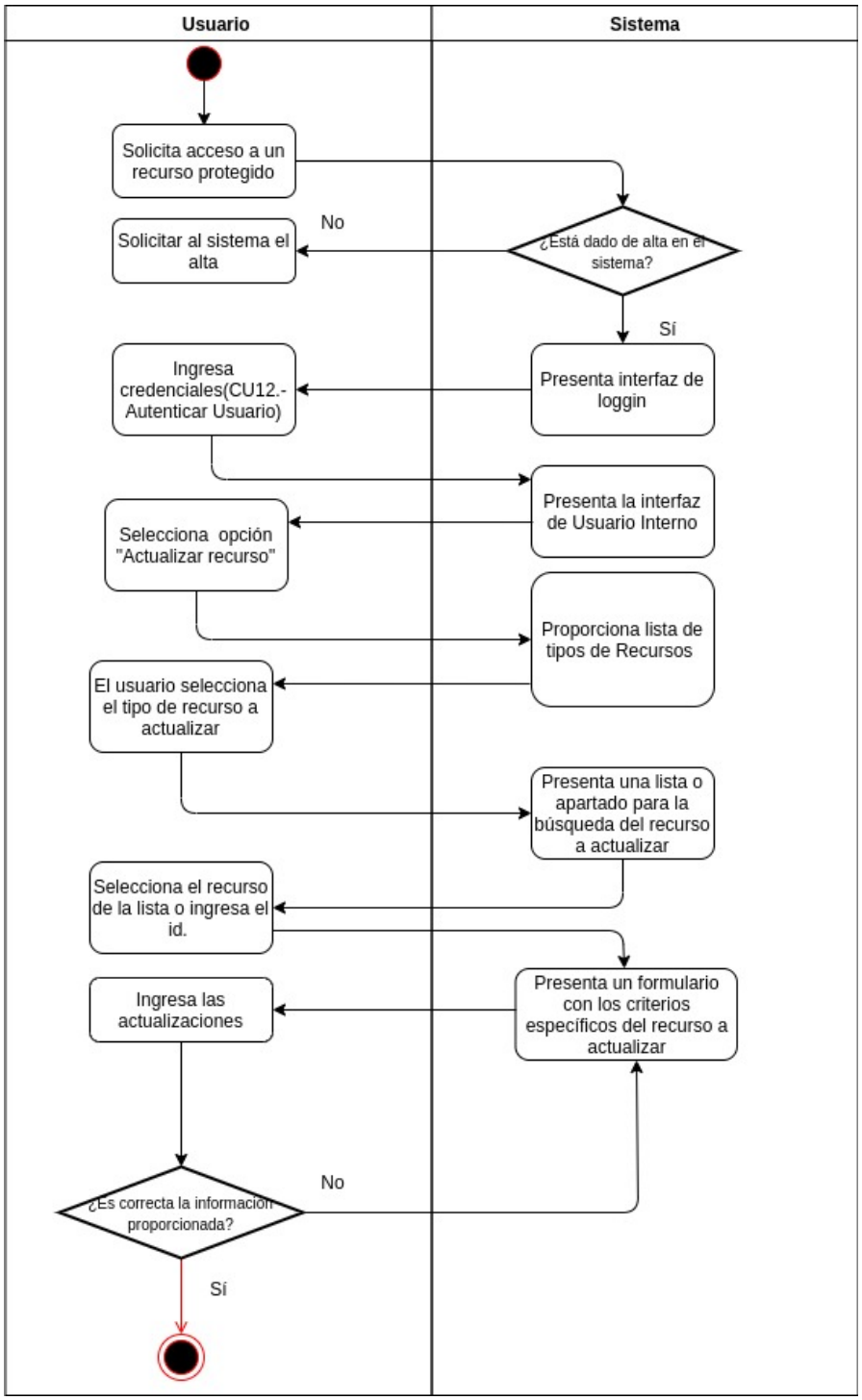


Figura 4.2.9 Diagrama de actividad Actualizar recurso

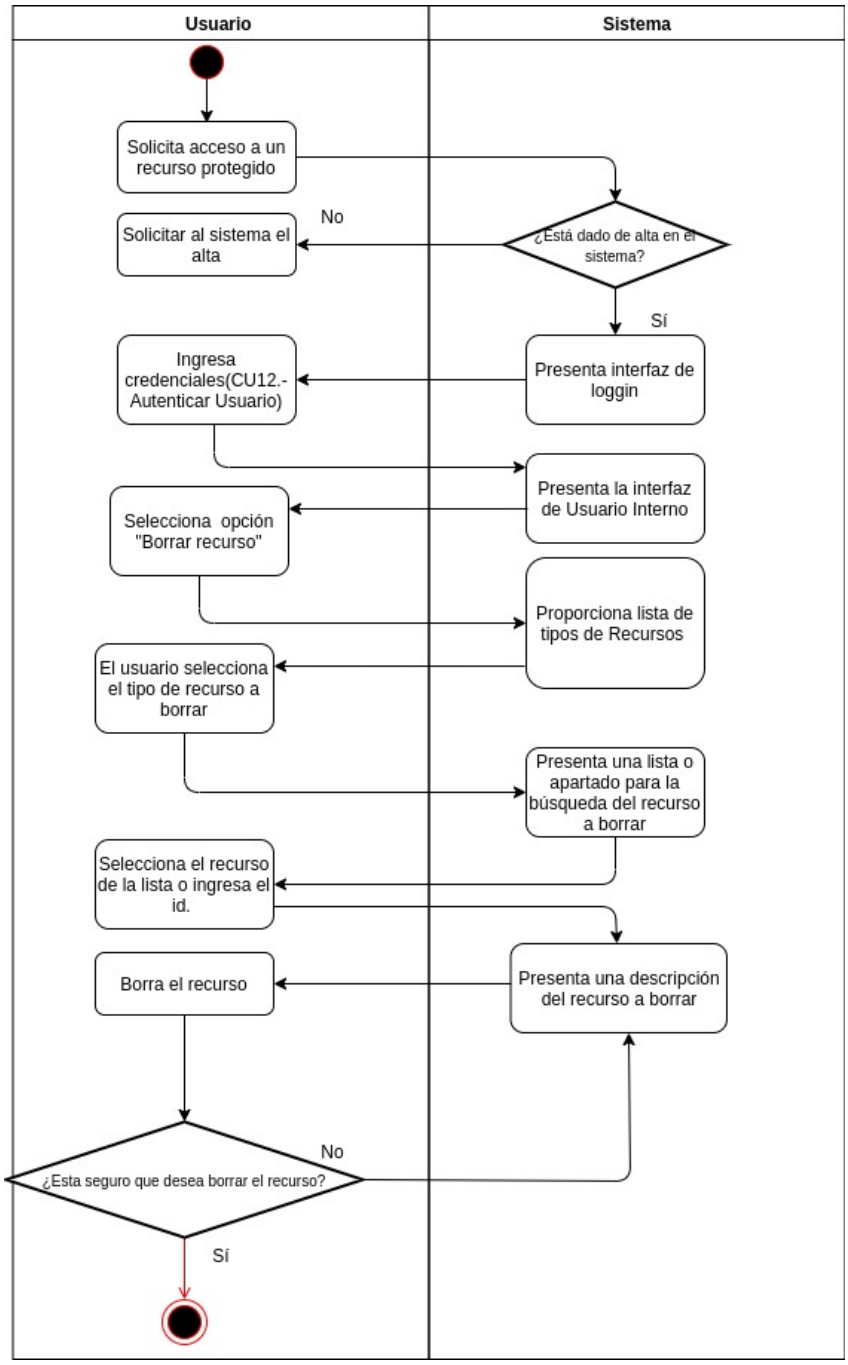


Figura 4.2.10 Diagrama de actividad Borrar recurso

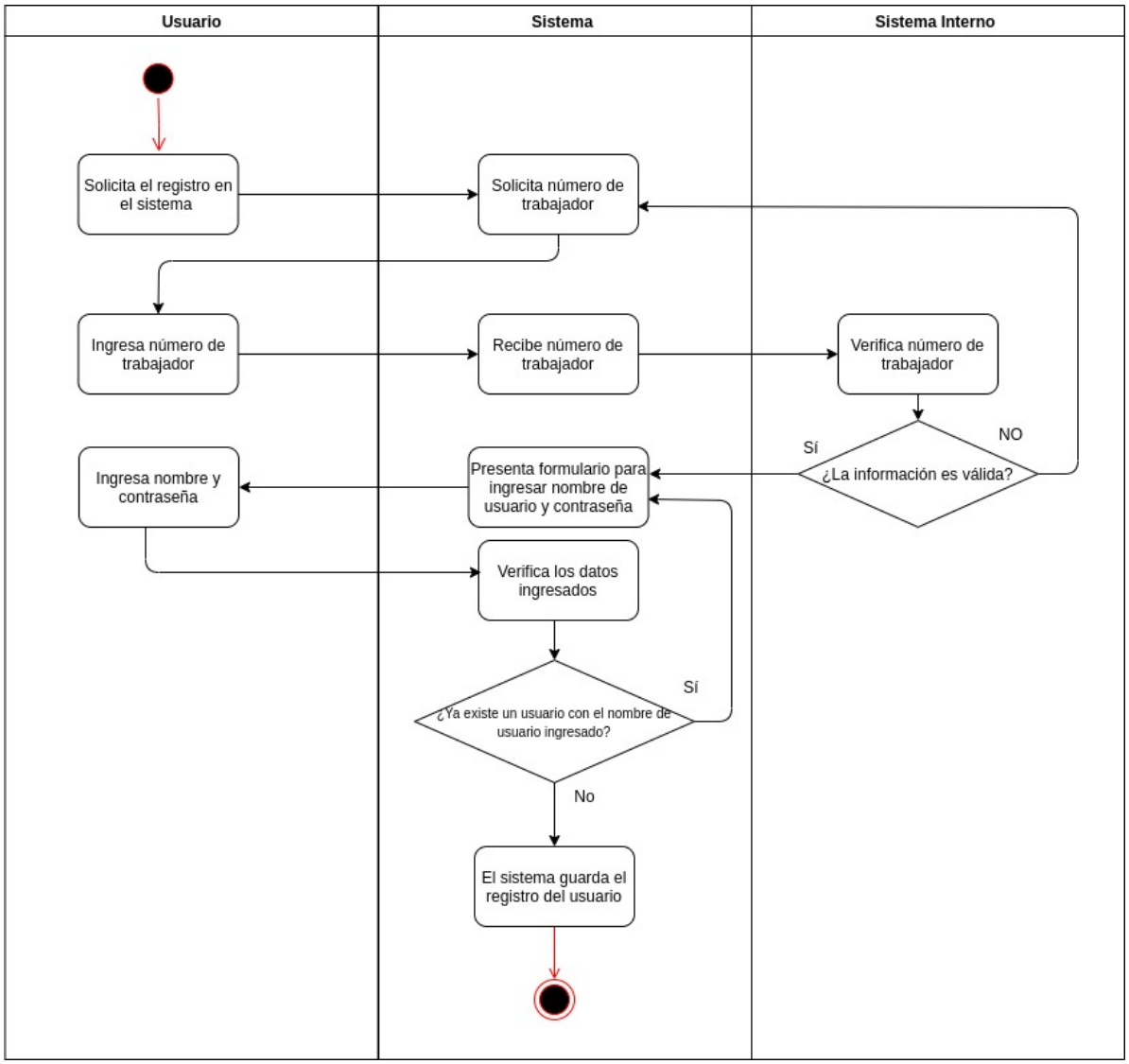


Figura 4.2.11 Diagrama de actividad Dar de Alta usuario

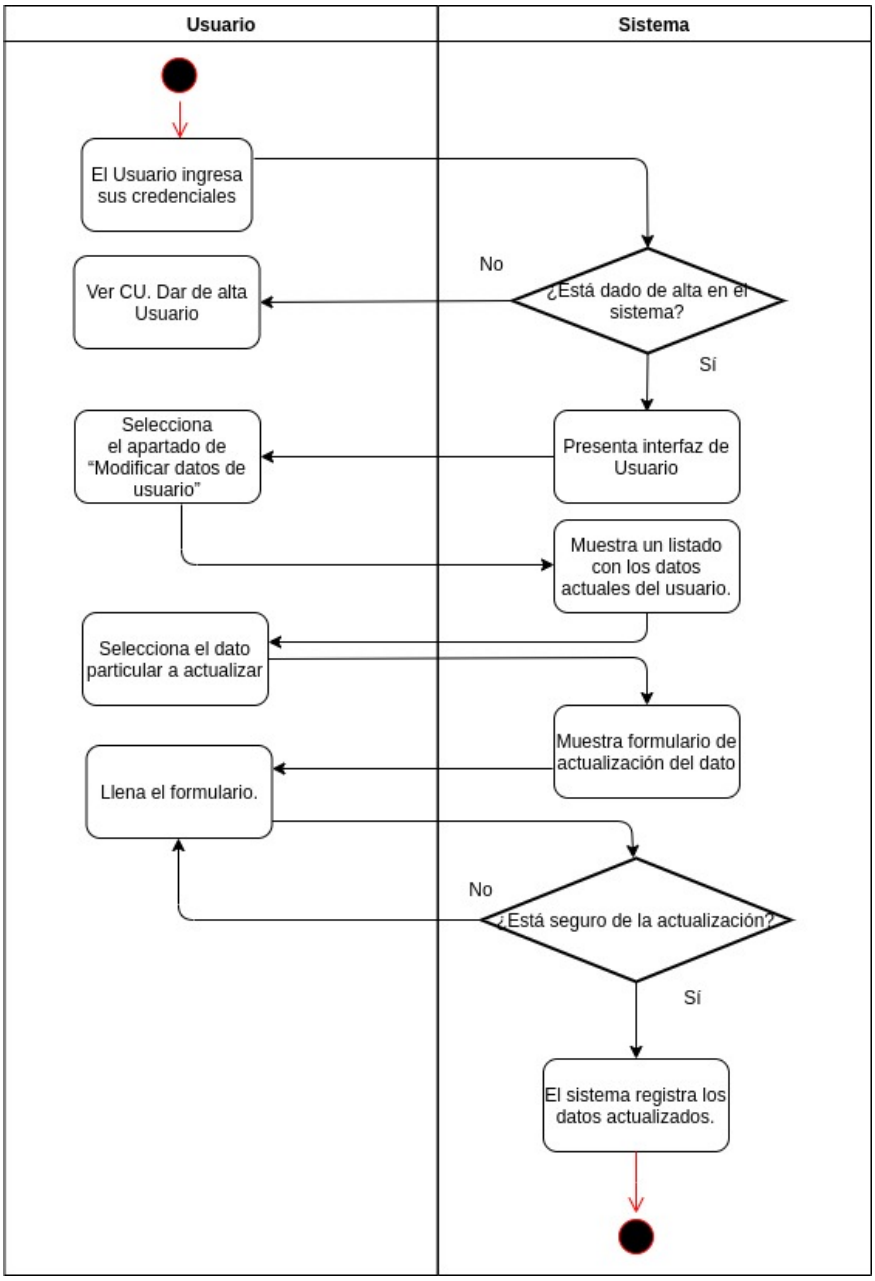


Figura 4.2.12 Diagrama de actividad Actualizar usuario

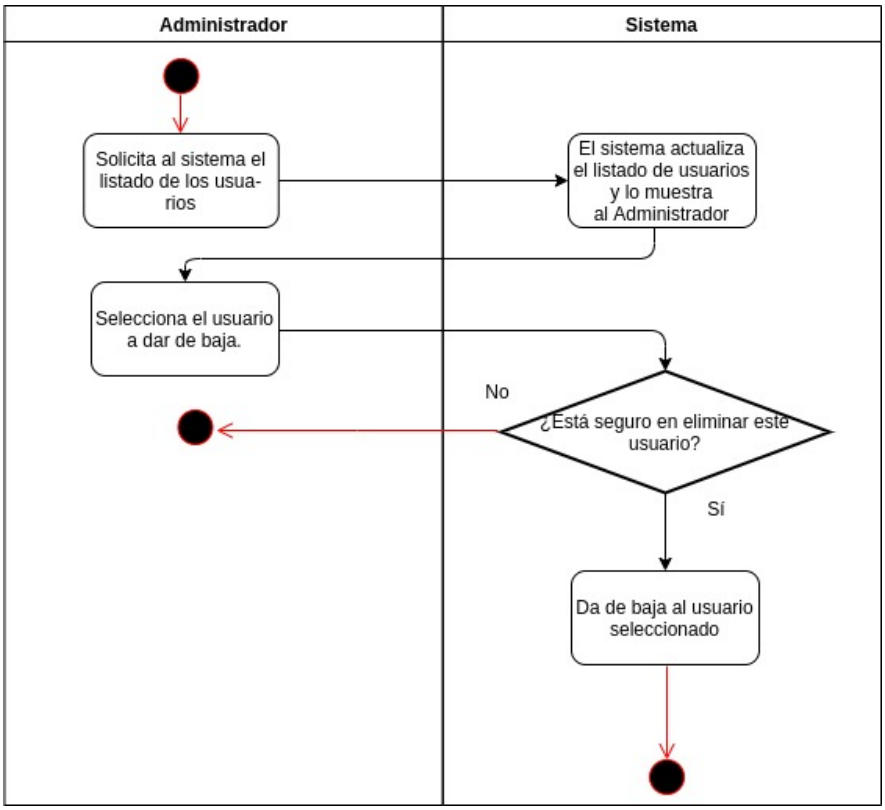


Figura 4.2.13 Diagrama de actividad Borrar usuario

4.2.2 Arquitectura

En cuanto a los componentes de software se utilizarán una serie de proyectos de código abierto que permitan el buen funcionamiento de la plataforma así como su fácil actualización. Ya que estos proyectos cuentan con comunidades grandes y activas en su desarrollo, por lo que se tienen actualizaciones frecuentes que apoyan el desarrollo de nuevas tecnologías, así como la incorporación de capas más seguras, que en caso de ser necesarias serán fáciles de aplicar.

Dado que durante el proceso de actualización del POETY se utilizará GRASS como sistema de información geográfica. Para la base de datos espacial se utilizará la combinación de GRASS con Postgres-PostGIS. Postgres es un manejador de bases de datos con capacidades espaciales en su módulo PostGIS. Adicionalmente se utilizará el manejador de metadatos Geonetwork que permitirá la transparencia en el manejo de todas las capas de información geográfica, los insumos que se usaron para crearlas y el procedimiento que se aplicó.

Para la lógica del sistema se utilizará Django, un marco de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como MVC (Modelo–Vista–Controlador). Una de las virtudes de Django es que responde de manera oportuna a las peticiones de los usuarios, aun cuando el sistema este teniendo una alta demanda. Como servidor http se utilizará Apache, Apache es un proyecto de código abierto muy consolidado (la primera versión se liberó en 1995). Finalmente, para desplegar

mapas y gráficas se utilizarán d3 y openlayers que son dos bibliotecas de código abierto programadas en JavaScript enfocadas en crear visualizaciones dinámicas con mapas y gráficas.

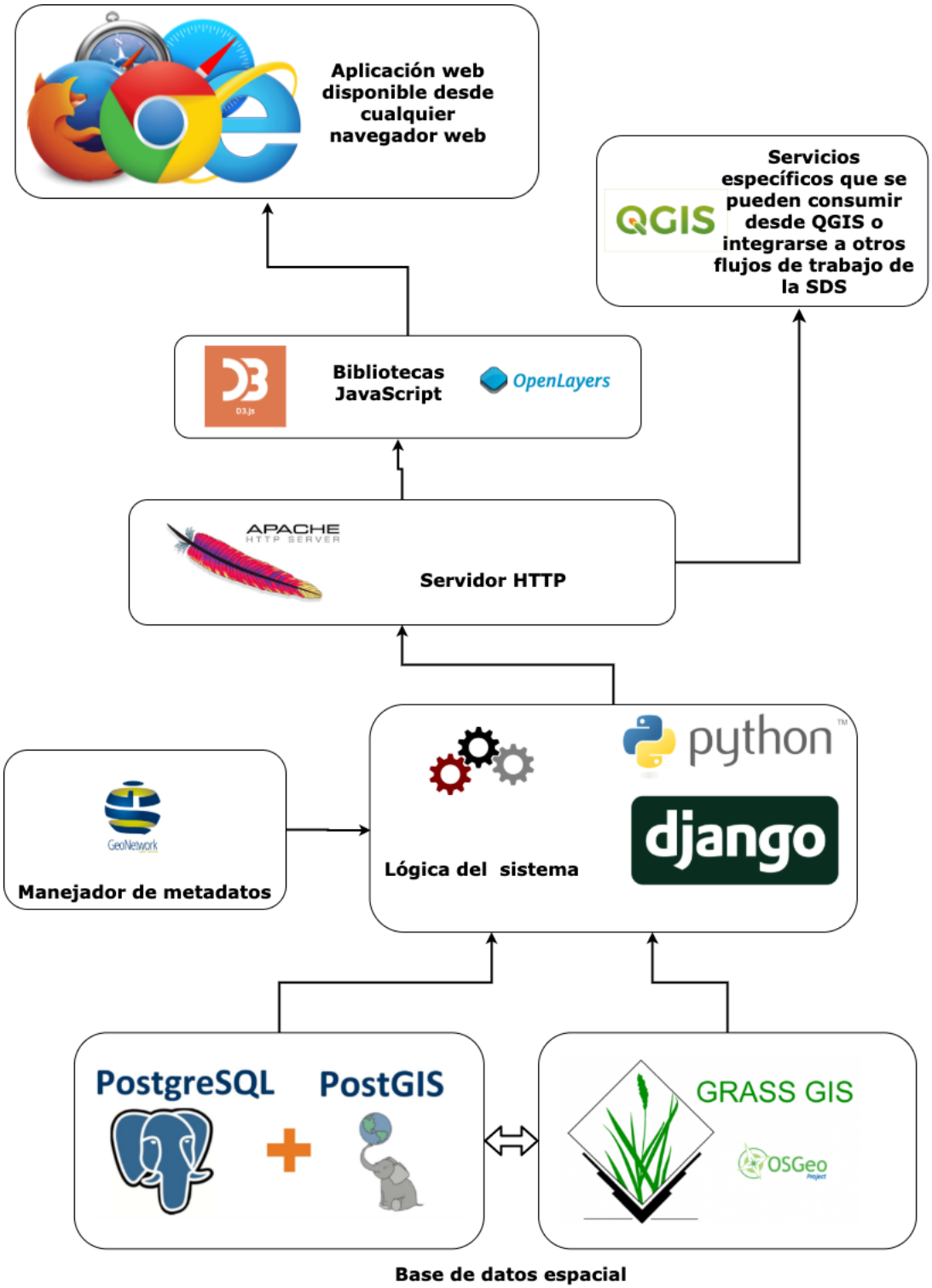


Figura 4.2.14 Componentes de software

El patrón MVC separa la lógica de la aplicación de la lógica de la vista, y de los bancos bancos de datos que soportan el sistema (el modelo). En nuestro caso la lógica de la aplicación será manejada por Django mientras que la parte del modelo se implementará con una combinación del manejador de bases de datos espaciales Postrgres-PostGIS, el sistema de información geográfica GRASS y el manejador de metadatos GeoNetwork. Finalmente, la vista será implementada con HTML y JavaScript minimizando las transferencias de datos para asegurar que las visualizaciones sean responsivas y veloces.

El sistema tendrá como principal vista un sitio web que se podrá consumir sede cualquier navegador. Adicionalmente tendrá servicios especificos como la consulta de criterios de regulación por polígono, que podrán ser consumidos desde QGIS o integrarse u otros flujos de trabajo de la SDS.

Anexo III

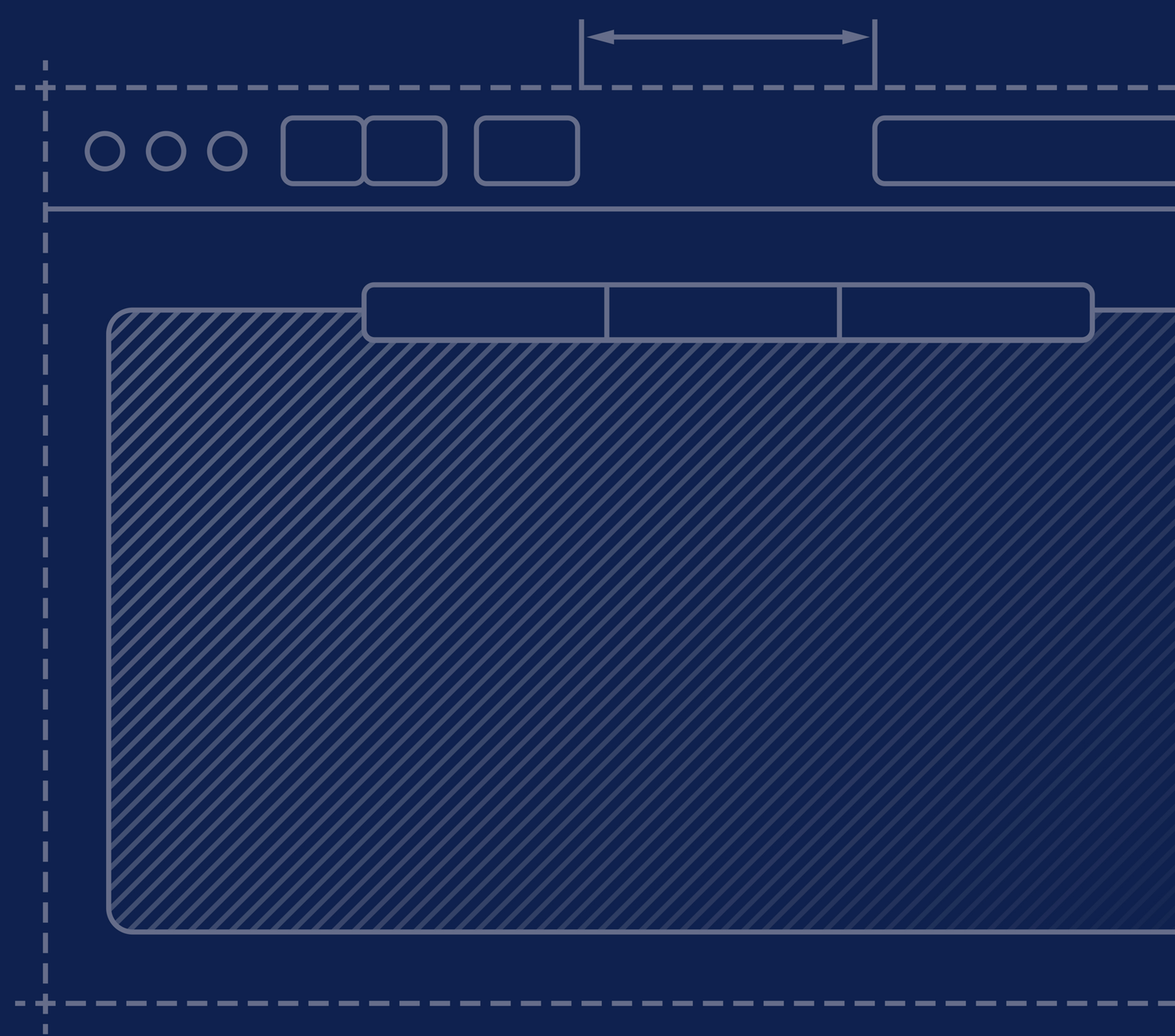
Guía de estilos para el diseño de páginas web

Juntos transformemos Yucatán
GOBIERNO ESTATAL 2018 · 2024

SAF
SECRETARÍA DE ADMINISTRACIÓN Y FINANZAS

YUCATAN

Guía de **estilos para plataformas**



G-PR-ACP -01 R00

GUÍA DE ESTILOS
para plataformas

PALETA DE COLORES

PALETA PRIMARIA

La paleta primaria puede ser implementar en: botones, iconos y títulos

#FFC78B	#B9E7F7
#FFA545	#45BFEB
#FF8400	#00A8E4

PALETA SECUNDARIA

Se puede implementar en textos

#EFF0F2	Color de fondo para Login #DDE2E9
#C4C9CE	#1C264E
#7E7E7E	#242F62

TIPOGRAFÍA

PANTON REGULAR	PANTON BOLD	PANTON BLACK
Gobierno del Estado	**Gobierno del Estado**	**Gobierno del Estado**

TEXTOS

TÍTULO 1 H1

PANTON BLACK 30 PX

ESPACIADO: 1%

TÍTULO 2 H2

PANTON BOLD 30 PX

ESPACIADO: 1%

TÍTULO 3 H3

Panton Bold 24 PX

ESPACIADO: 1%

TEXTO DESCRIPTIVO

Panton Regular 18 px
Para Textos principales

El Veloz Murciélago Hindú Comía Feliz Cardillo Y Kiwi. La Cigüeña Tocaba El Saxofón Detrás Del Palenque De Paja

Panton Regular 16 px
Para Textos complementarios

El Veloz Murciélago Hindú Comía Feliz Cardillo Y Kiwi. La Cigüeña Tocaba El Saxofón Detrás Del Palenque De Paja

BOTONES

Estilo de botón normal

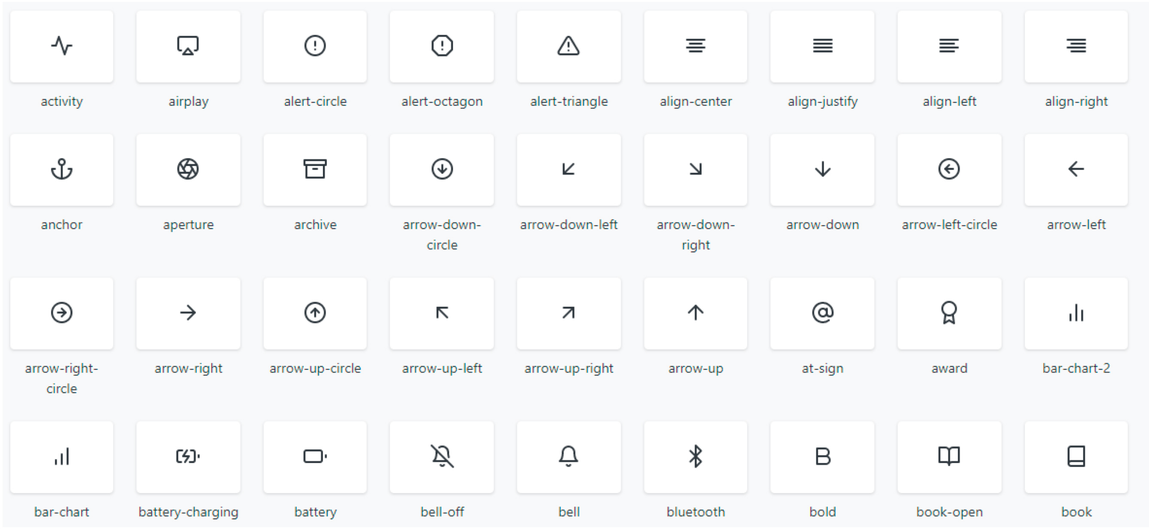
Normal	Hover	Activo	No Disponible
Normal	Hover	Activo	No Disponible

Estilo de botón de cancelar

Normal	Hover	Activo	No Disponible
Normal	Hover	Activo	No Disponible

ICONOS

feathericons.com https://github.com/feathericons/feather



INPUT DE FORMULARIO

Campo de texto

Hover/Select

Texto de ejemplo

Campo de texto

Aprobado

Campo de texto

Texto de error o verificación

Select

Seleccione una opción

Select activo

Texto de ejemplo

Campo bloqueado

Campo de texto

CHECKBOXES

Texto Texto Texto

Texto Texto Texto

SCROLL

RECAPTCHA

No soy un robot

reCAPTCHA

Privacidad - Condiciones

FOOTER

Home

Juntos transformemos Yucatán

GOBIERNO ESTATAL 2018 - 2024

DIRECTORIO DIARIO OFICIAL TRANSPARENCIA

Emergencia 911

Denuncia Anónima 089

GOBIERNO DEL ESTADO DE YUCATÁN 2018-2024

Hoja interna

Juntos transformemos Yucatán

GOBIERNO ESTATAL 2018 - 2024

Subsecretaría de Adquisiciones

Dirección: Calle 59 por Av. Itzáes y calle 90, Centro, C.P. 97000, Mérida, Yucatán
Teléfono: (999) 930 33 40 / Extensión: 57017
Correo: padron.proveedores@yucatan.gob.mx

Emergencia 911

Denuncia Anónima 089

GOBIERNO DEL ESTADO DE YUCATÁN 2018-2024

CARDS

Datos de pago

Nombre

Nombre

Apellido

Apellido

Activo

TABLAS

Título de la tabla

Primero	Segundo	Tercero
Primero	Segundo	Tercero
Primero	Segundo	Tercero
Primero	Segundo	Tercero
Primero	Segundo	Tercero

MODAL

Título

×

El Veloz Murciélago Hindú Comía Feliz Cardillo Y Kiwi. La Cigüeña Tocaba El Saxofón Detrás Del Palenque De Paja

Cerrar

Gobierno del Estado de Yucatán 2018-2024
Calle 61 x 60 y 62 Col. Centro C.P. 97000 Mérida, Yucatán, México
Tel. (999) 930 31 00

Subsecretaría de Tecnologías de la Información y Comunicaciones

mesadeayuda@yucatan.gob.mx

Tel. 4.00.02.99

Anexo IV
Lineamientos de trabajo web

Juntos transformemos Yucatán
GOBIERNO ESTATAL 2018 · 2024

SAF
SECRETARÍA DE ADMINISTRACIÓN Y FINANZAS

YUCATAN

LINEAMIENTOS DE TRABAJO WEB

CONTENIDO

CONTENIDO 2
INTRODUCCIÓN 4
SEGURIDAD WEB 5
CODIFICACIÓN 6

CODIFICACIÓN FISICA DE LOS ARCHIVOS FUENTE 6

UTF-8 (SIN BOM) (Importante) 6
Fin de Linea (LF) (Deseable) 6

RECOMENDACIONES DE PROGRAMACIÓN 6

HTML y CSS 6
Nombres de archivos HTML 6

RESPONSIVIDAD 7

Librerías CSS para responsividad 7

RECOMENDACIONES GENERALES DE HTML5 8

RECOMENDACIONES GENERALES DE CSS 9

Nombres de archivos CSS 9
Codificación 9
Modos de uso CSS 12

LENGUAJES DE PROGRAMACION WEB 12

Nombres de archivos de código 13
Codificación 13
Estructura de carpetas y códigos WEB 14

HERRAMIENTAS PARA MANEJO DE PAQUETES 16

COMPOSER PARA PHP 16

Cómo utilizarlo 16
CLI de composer 18
Instalación de paquetes 18
Desinstalación de paquetes 19
Instalación y actualización de paquetes incluidos en composer.json 20

BOWER PARA JAVASCRIPT ES5 20

Requerimientos previos 21
Instalación 21
Instalación de paquetes 21
Desinstalar paquetes 23
Actualizar paquetes 23

NPM PARA FRONTENDS ES6 O BACKEND NODEJS 23
Instalación 24
Instalación de paquetes 24
Desinstalar paquetes 26
Actualizar paquetes 26
Capacidades de ES6 26
REPOSITORIOS DE CÓDIGO 28
GITLAB 28
Proyectos de GitLab 28
Proyectos de usuario o grupo 30
Opciones de creación de proyectos 30
Grupos 30
Usuarios 31
Roles de trabajo 31

INTRODUCCIÓN

Las nuevas implementaciones de trabajos web deben cumplir requerimientos que permitan mejorar la calidad de software. Esto también sienta las bases para actualizar componentes WEB antiguos a un nuevo modelo de trabajo, las consideraciones principales son:

- Seguridad WEB
- Codificación
- Guías de codificación
- Herramientas para manejo de paquetes
- Repositorios de código

Nota importante: Al momento de la publicación todos los recursos que se toman del repositorio de Gobierno requieren acceso explícito al repositorio; esta disposición está sujeta a cambiar previa revisión.

SEGURIDAD WEB

El servidor en el que se alojan los sitios web de las dependencias es compartido, y cualquier vulnerabilidad de seguridad en un sitio podría comprometer la seguridad en los demás sitios alojados en el mismo servidor, por lo que es necesario cumplir con las siguientes recomendaciones en cualquier sitio publicado bajo el dominio **yucatan.gob.mx**:

1. El código debe tener todas las validaciones necesarias para evitar la inyección SQL.
2. Limpiar todos los datos provenientes de formularios o de la barra de direcciones: eliminar palabras reservadas como `create, insert, update, delete, drop, select, union, alter, grant, truncate, between, or, jquery, javascript, script`; eliminar caracteres especiales como `< > % $ ( ) " '` o en su defecto, convertirlos a sus equivalentes códigos HTML.
3. Validar el código HTML ingresado por el usuario por medio de algún editor de texto enriquecido (TinyMCE, CKEditor, etc.), en especial los estilos en línea, por ejemplo, `<DIV STYLE="background-image: url(javascript:alert('XSS'))">`.
4. Validar que las variables acepten únicamente los tipos de datos necesarios (por ejemplo, una variable numérica no debe aceptar letras).
5. Si el sitio acepta archivos como entrada, limitar el tipo de archivos que se pueden enviar, y no aceptar archivos ejecutables o posiblemente maliciosos.
6. No mostrar mensajes de error que contengan información de la estructura de la base de datos, la instrucción SQL que generó el error, el tipo de servidor que se utiliza, etc.
7. Utilizar CAPTCHAS en todos los formularios.
8. En caso tener módulos que requieran autenticación, las contraseñas aceptadas deben ser robustas y libres de ataques por fuerza bruta:
    - Deben almacenarse encriptadas en la base de datos (de preferencia, con el algoritmo SHA-1).
    - Tener una longitud mayor a 8 caracteres.
    - Utilizar una combinación de letras mayúsculas y minúsculas, números y caracteres especiales.

CODIFICACIÓN

CODIFICACIÓN FISICA DE LOS ARCHIVOS FUENTE

UTF-8 (SIN BOM) (IMPORTANTE)

Este formato de codificación de caracteres está catalogado como una regla y se espera que sea aplicado a todos los archivos de código fuente que conforman la página o aplicación, de igual forma se recomienda que en caso de utilizar bases de datos que permitan implementar codificación, se utilice UTF-8. Ejemplo: mysql (utf8_default_collation).

Importancia de remover el BOM

Aunque hoy en día los principales IDEs para desarrollo web consideran su eliminación por defecto (VSCode, Meteor, Web/PhpStorm) existen algunos como el Notepad++ que dan la opción de incluir el BOM en el formato UTF-8, lo que presenta riesgo de inconsistencias y problemas de visualización en la salida de la información.

FIN DE LINEA (LF) (DESEABLE)

Se sugiere utilizar el fin de línea de UNIX en los códigos fuente, es decir, el uso del carácter LF a diferencia del CRLF, ya que no se requiere realmente el carácter adicional que incluye el fin de línea de estilo Windows CRLF

RECOMENDACIONES DE PROGRAMACIÓN

HTML Y CSS

En cuanto a HTML, la recomendación es dejar de utilizar estándares antiguos tales como HTML4 o XHTML, el estándar definido y recomendado es el HTML5.

NOMBRES DE ARCHIVOS HTML

La recomendación en cuanto a los estándares de nomenclatura para los nombres de archivos HTML, es el uso de snake case, es decir, separar cada una de las palabras por un guion bajo (_), ejemplos:

- **index.html**
- **formulario_contacto.html**
- **not_found.html**

Cómo indicar al navegador que utilizas HTML5 (DOCTYPE)

XHTML (Forma incorrecta de codificar)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

HTML5 (Forma correcta)

```
<!DOCTYPE html>
```

Parece algo sin importancia, pero con esta instrucción HTML se le indica al navegador qué características podría o no utilizar, hoy en día la mayoría de los navegadores son extremadamente sensibles a esto.

RESPONSIVIDAD

Actualmente, el 52% de la navegación web mundial se hace a través de dispositivos móviles, razón por la cual los diseños responsivos son una **regla** interna para los sitios y web apps.

Primeros pasos de un sitio responsivo

incluir el **meta tag** correspondiente para manejo de escalas de pantalla (**viewport**):

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
```

Nota: Este ejemplo incluye además el meta para forzar al navegador a interpretar el contenido como UTF-8

LIBRERÍAS CSS PARA DISEÑO WEB RESPONSIVO

Actualmente existen diversas librerías CSS que permiten utilizar de manera sencilla el diseño web responsivo. Aquí algunas recomendaciones:

- Bootstrap – Recomendado
- Bulma - Recomendado
- Foundation
- Skelleton

Nota: En pro de la innovación y mejora dejamos elecciones abiertas para el manejo de hojas de estilos, sin embargo, es importante recalcar que se debe elegir un Framework CSS que permita cumplir con las exigencias de estilos visuales requeridas por la entidad a cargo, así que en su elección considere uno que pueda escalarse y adaptarse de manera correcta.

RECOMENDACIONES GENERALES DE HTML5

1. Ya no es necesario indicar el parámetro **type** en etiquetas **link** o script.

- Incorrecto

```
<link rel="stylesheet" type="text/css" href="style.css "/>
<script type="text/javascript" src="script.js"></script>
```

- Correcto

```
<link rel="stylesheet" href="style.css">
<script src="script.js"></script>
```

2. No necesitas indicar la diagonal '**/**' para cierre de tags simples.

- Incorrecto

```
<br />
<input type="text" id="text" />
```

- Correcto

```
<br>
<input type="text" id="text">
```

3. Se recomienda aprovechar las nuevas funciones de input y evitar la carga de scripts para agregar funcionalidades que ahora vienen por default en los navegadores, como campos de número o email.

```
<input type="date" name="fecha" id="fecha">
<input type="time" name="hora" id="hora">
<input type="datetime-local" name="fecha-hora" id="fecha-hora">
<input type="number" name="numero" id="numero" min="1" max="10">
<input type="url" name="url" id="url">
<input type="range" name="rango" id="rango">
<input type="email" name="correo " id="correo">
```

4. Como regla tenemos el no utilizar estilos con tags depreciados que han sido suplantados por hojas de estilo css. Los más comunes que hemos encontrado son:

```
<bold>
<font>
<center>
```

5. No utilizar layouts basados en *frames* o *tables*; éstos están depreciados desde HTML4 y entran en conflicto directo con la regla de responsividad.

Esto no quiere decir que no podamos presentar información en formato de tablas o utilizar iframes para mostrar contenido externo, sin embargo, recomendamos hacer uso de ello en la menor medida posible.

Para una lista más completa y verificación de compatibilidad entre navegadores, sugerimos usar el recurso online: Can I Use

RECOMENDACIONES GENERALES DE HOJAS DE ESTILO CSS

NOMBRES DE ARCHIVOS CSS

La recomendación de estándar para la nomenclatura de los nombres de archivo CSS es el uso de kebab, es decir, separar cada una de las palabras por un guion medio (-), ejemplos:

- **style.css**
- **image-slider.css**
- **shadow-image-box.css**

CODIFICACIÓN

Para los nombres de las clases CSS la regla es utilizar la notación **BEM** (Block Element Modifier), ya que lo hace más entendible para su uso en la semántica html, ejemplo:

- Incorrecto

```
.galContainer{
    padding: 12px;
}
```

- Correcto

```
.gal-container{
    padding: 12px;
}
```

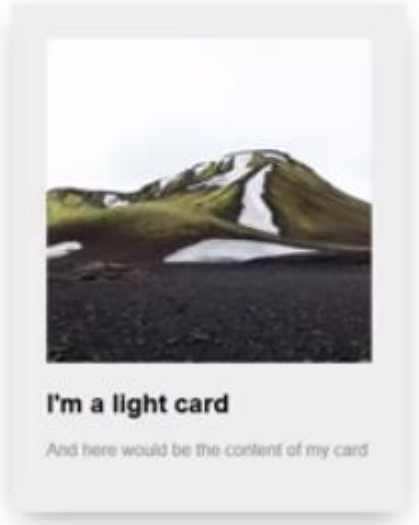
A manera de recomendación y sugerencia tenemos el adoptar la metodología BEM de forma integral en sus estilos css, utilizando clases y modificadores según la función que les corresponde y evitando anidamientos, ejemplo:

Una forma común (incorrecta en BEM) de realizar un estilo para un card seria este:

```
<!-- EL HTML -->
<div clas="card">
    <img src="logo.png" alt="logo">
    <h2>I'm a light card</h2>
    <p>Lorem ipsum</p>
```

```
</div>
/*EL CSS*/
.card {
    width: 300px;
    height: 400px;
    padding: .5em;
    border: 1px solid #000;
    margin: 0 1em;
    color: #0f0f0f;
}
.card p {
    font-weight: bold;
}
.card h2 {
    font-weight: bold;
    font-size: 1.5em;
}
.card img {
    margin: 1em;
    width: 100%;
    height: auto;
}
```

- Resultado



Si bien el resultado es el esperado, la forma tradicional nos está limitando la capacidad de hacer modificaciones al card principal. BEM nos sugiere el uso de modificadores y clases específicas, ejemplo:

```
<!--EL HTML-->
<div clas="card">
    <img class="card__image" src="logo.png" alt="logo">
    <h2 class="card__title">Titulo</h2>
    <p class="card__text">Lorem ipsum</p>
</div>
/*EL CSS*/
.card {
    width: 300px;
    height: 400px;
    padding: .5em;
    border: 1px solid #000;
    margin: 0 1em;
    color: #0f0f0f;
}
.card__text {
    font-weight: bold;
}
.card__title {
    font-weight: bold;
    font-size: 1.5em;
}
.card__image {
    margin: 1em;
    width: 100%;
    height: auto;
}
```

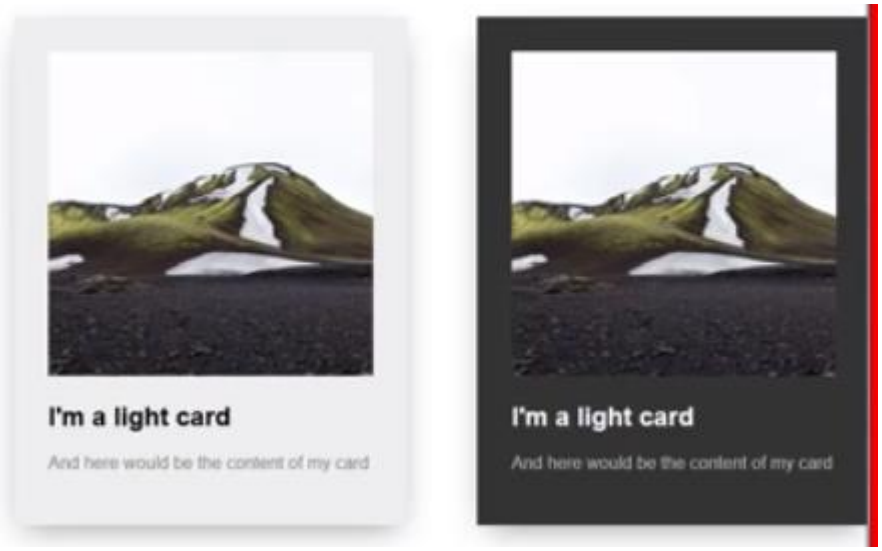
El resultado es el mismo, pero nuestra capacidad de modificar se expande con menos código. Digamos que queremos cards de diversos colores:

```
<!-- Agregamos un modificador de color 'card--dark'-->
<div clas="card card--dark">
    <img class="card__image" src="logo.png" alt="logo">
    …
/*-Definimos el modificador en el codigo CSS-*/
.card--dark {
    background: #0f0f0f;
    color: #ddd;
```

```
}
```

Al final nuestros estilos serán más moldeables y entendibles ya que desde el HTML podremos observar las clases independientes de cada elemento y no tendremos que preocuparnos si en el CSS hay algún tipo de animamiento que no estemos contemplando.

- Resultado



MODOS DE USO CSS

- Hoja de estilos (**recomendado**) - Deben ir dentro del <head>

```
<link rel="stylesheet" href="ruta/style.css">
```

- Etiqueta Style - Deben ir preferentemente en el <head> y por debajo de las hojas de estilo, solo se recomienda en páginas independientes.

```
<style> .gal-container{ padding: 12px; } </style>
```

- CSS Inline - Solo en casos que requieran su uso explícito

```
<div style="width: 100%"></div>
```

LENGUAJES DE PROGRAMACION WEB

Las siguientes recomendaciones aplican para los lenguajes web comúnmente utilizados (Javascript, PHP, C# .Net, Ruby on Rails, etc.)

NOMBRES DE ARCHIVOS DE CÓDIGO

Se recomienda para los lenguajes de programación web el uso de Camel Case (la primera letra de cada una de las palabras es mayúscula, con excepción de la primera), permitiendo el uso de mayúscula inicial para Clases o Componentes, ejemplo:

- **main.js**
- **apiGobierno.js**
- **routes.php**
- **GobiernoController.php** (clase)
- **TopNavBar.js** (componente)

CODIFICACIÓN

Al igual que en los nombres de archivos, la codificación recomendada es mediante el uso de Camel Case, minúscula para funciones y variables, mayúscula para clases, exceptuando constantes que se recomiendan en snake case y mayúsculas:

```
/** variables y funciones **/
function fooBar() {
  var logVar = 123;
  console.log(logVar);
}

/** Clases **/
class GobiernoController() {
  ...
}

/** constantes **/

const API_GOBIERNO_KEY = '2342i34oi2h219834n92n498213j4=='
```

Para mayor información sobre recomendaciones web puedes utilizar nuestro recurso de documentación más completo en el siguiente repositorio: [Documentación - Recomendaciones de programación Web]

ESTRUCTURA DE CARPETAS Y CÓDIGOS WEB

La manera de cómo se estructura y se maneja el código fuente de un proyecto es responsabilidad de cada desarrollador, sin embargo, aquí se provee una estructura sugerida, así como una template de proyecto base y documentación adicional requerida. Los ejemplos son considerados para código con un backend PHP y un frontend abierto, pero pueden adaptarse a otro tipo de lenguajes.

Estructura sugerida

- App - Carpeta para guardar y estructurar el código backend de nuestra aplicación, estructurando de manera correcta las clases y funciones de ayuda. Un ejemplo de estructura interna:
    - Clases
    - Templates
    - Catalogos
- Config - Carpeta para incluir todos los archivos de configuración requeridos (variables de entorno o configuraciones a BD).
- Files-Content - Carpeta para manejo de archivos estáticos, manejando una estructura interna que convenga para mantener el orden. Ejemplo de estructura interna:
    - Uploads
    - Cache
    - Documentos
- vendor - En caso específico de PHP se utiliza para almacenar librerías de terceros.
- Public o App - Carpeta con archivos de la parte pública (frontend) de la aplicación. La estructura recomendada para el frontend es la siguiente:
    - css - Carpeta para estilos
    - img - Imágenes estáticas del proyecto
    - js - Carpeta para scripts propios
    - fonts - Carpeta para fuentes utilizadas en frontend
    - favicon - puede ser un archivo **ico** o un folder con estructura de favicons para dispositivos diversos.

- lib - Carpeta para librerias de terceros

Nota: El caso de nombrar la carpeta de Public como App o semejante es recomendado únicamente para aplicaciones compiladas, como por ejemplo Apps de Angular, Vue, React, etc., ya que es un mejor indicativo del contenido y de que la estructura interna del código puede variar de acuerdo al tipo de App.

Actualmente tenemos un Boilerplate PHP listo para uso y con métodos básicos de trabajo para desarrollo de aplicaciones WEB. Puedes usar las siguientes ligas de repositorio:

- DAI PHP Backend - Proyecto de App base para desarrollo
- Presentación de slides del proyecto
- Documento de los slides del proyecto

HERRAMIENTAS PARA MANEJO DE PAQUETES

Algunos lenguajes como Ruby on Rails cuentan con su propio manejador de paquetes (GEM). Para PHP o Javascript también existen y es una recomendación utilizarlos por las siguientes ventajas que ofrecen:

- Permiten controlar la versión de la librería que deseas incluir en tu proyecto
- Permiten actualizar las librerías del proyecto y realizar cambios de versión sin necesidad de cambiar código
- Permiten reducir el tamaño del código fuente ya que no es necesario guardar las librerías dentro del proyecto, sino que pueden instalarse cuando se requiera.
- Permiten trabajar en diversos ambientes de ejecución sin conflictos.

COMPOSER PARA PHP

Es un manejador de paquetes para PHP que proporciona un estándar para administrar, descargar e instalar dependencias y librerías. Prácticamente, cualquier librería de terceros libre que encontremos en páginas como GitHub, están disponibles en su lista de paquetes y pueden ser manejadas en composer; muchas incluso, solo recomiendan instalarse mediante este manejador y no intentarlo de manera manual.

- Sitio principal y de descarga de composer
- Packagist - Lista general de repositorio de paquetes PHP

CÓMO UTILIZARLO

Una vez instalado, podemos configurarlo en nuestro proyecto incluyendo un archivo composer.json en la raíz de nuestro proyecto php.

El Boilerplate sugerido en apartados anteriores maneja sus librerías por este método.

Este archivo contiene algo de información sobre nuestro proyecto, versión de PHP requerida, listado de paquetes necesarios, entre otros.

Si no se especifica la versión de PHP, ésta dependerá de nuestra versión de PHP de sistema, es decir, no puedes indicar que tu proyecto utiliza una versión PHP 5.6 si tu instalación es una versión menor, sin embargo, no habría problemas si tu versión fuese mayor.

La versión de PHP es importante en el sentido que composer instalará las versiones de librerías específicas que sean compatibles con la versión de PHP que indiquemos o esté instalada.

Ejemplo de composer.json:

```
{
  "name": "DAI Backend",
  "type": "Web Backend",
  "description": "Boilerplate para inclusión del backend DAI en
tus proyectos",
  "license": "MIT",
  "support": {
      "docs": "https://gitlab.yucatan.gob.mx/DAI/dai-
backend/master/readme.md",
      "issues": "https://gitlab.yucatan.gob.mx/DAI/dai-
backend/issues",
  }
  "config": {
      "platform": {
          "php": "5.6.39"
      }
  },
  "require": {
    "illuminate/database": "*",
    "slim/slim": "3.*",
    "phpmailer/phpmailer": "^5.2",
    "cboden/ratchet": "0.3.*",
    "doctrine/dbal": "*",
    "slim/twig-view": "^2.4"
  },
  "require-dev": {

  }
}
```

El apartado **require** es el más importante. Aquí podemos incluir la lista los paquetes que necesitamos instalar. Éstos también se insertan de manera automática en el listado si se instalan en el proyecto por CLI de composer.

El apartado **require-dev** es similar, pero solo sirve para librerías que únicamente utilizamos durante el desarrollo pero no son necesarias para el trabajo en producción.

CLI DE COMPOSER

La documentación completa del manejo del CLI puede encontrarse en los docs oficiales de composer; Composer CLI DOCS, en este apartado únicamente sugerimos el uso básico que comprende:

- Instalación de paquetes
- Desinstalación de paquetes
- Instalación y actualización de paquetes listados en composer.json

Nota: Para Windows, asegúrate de haber hecho que la ruta a composer esté incluida en el PATH de sistema, de lo contrario no podrás ejecutar los comandos.

INSTALACIÓN DE PAQUETES

La instalación inicial de paquetes únicamente requiere que tengas el nombre del paquete listo para uso. A la fecha, la mayoría de las librerías de terceros incluyen el dato en su documentación inicial; de no hacerlo, la convención estándar es utilizar su nombre en Github. Por ejemplo, para PHP Mailer:

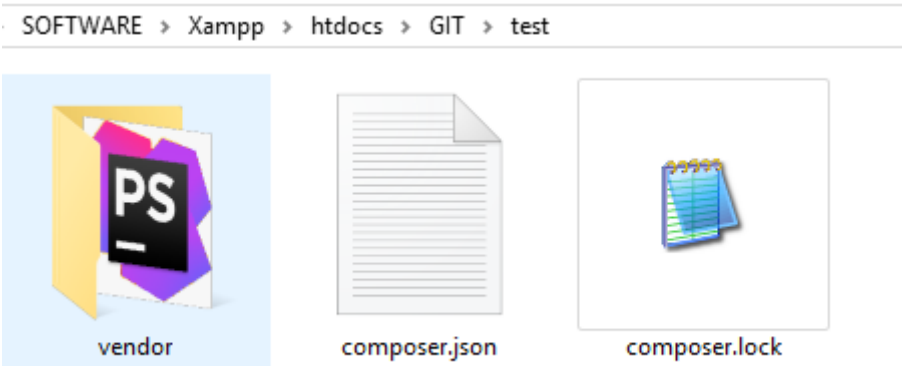
URL de GitHub `https://github.com/PHPMailer/PHPMailer`

Por lo tanto, el nombre del paquete es `PHPMailer/PHPMailer`

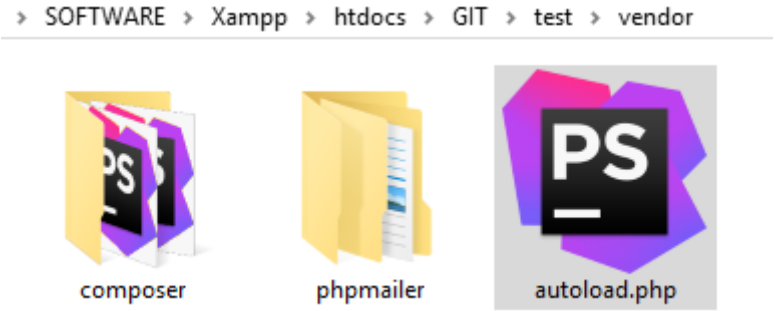
Comando de instalación composer `require PHPMailer/PHPMailer`, ejemplo:

```
Windows PowerShell
PS D:\SOFTWARE\Xampp\htdocs\GIT\test> composer require PHPMailer/PHPMailer
Using version ^6.0 for phpmailer/phpmailer
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing phpmailer/phpmailer (v6.0.7): Downloading (100%)
phpmailer/phpmailer suggests installing psr/log (For optional PSR-3 debug logging)
phpmailer/phpmailer suggests installing league/oauth2-google (Needed for Google XOAUTH2 authentication)
phpmailer/phpmailer suggests installing hayageek/oauth2-yahoo (Needed for Yahoo XOAUTH2 authentication)
phpmailer/phpmailer suggests installing stevenmaguire/oauth2-microsoft (Needed for Microsoft XOAUTH2 authentication)
phpmailer/phpmailer suggests installing symfony/polyfill-mbstring (To support UTF-8 if the Mbstring PHP extension is not
 enabled (^1.2))
Writing lock file
Generating autoload files
PS D:\SOFTWARE\Xampp\htdocs\GIT\test>
```

Una vez completa la instalación, se habrá generado automáticamente en la raíz una carpeta **vendor**, y en caso de no existir, los archivos **composer.json** y **composer.lock** con la librería requerida y sus dependencias.



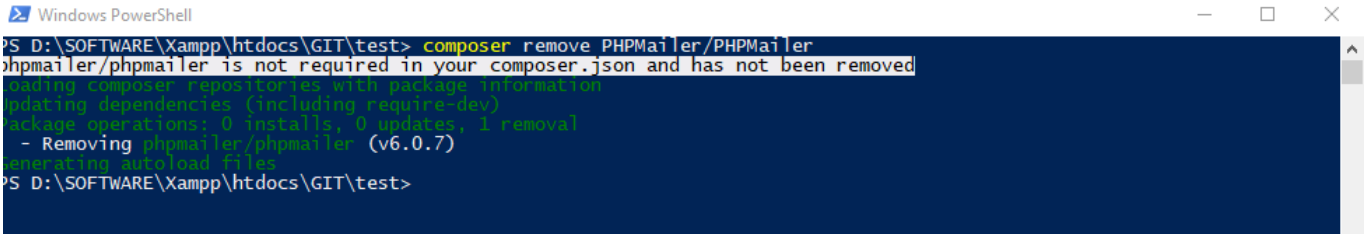
Este proceso se puede repetir para todas las librerías de terceros que nuestro proyecto requiera, y al final otra ventaja de composer es que nos da un punto único de inclusión. El archivo **vendor\autoload.php** es el único archivo al que tenemos que hacer referencia en el código (include o require) y éste se encarga de cargar las librerías requeridas bajo demanda automática.



DESINSTALACIÓN DE PAQUETES

Este proceso es requerido cuando deseamos remover una dependencia del proyecto. No podemos simplemente borrar la carpeta, ya que la referencia al paquete permanecería en **composer.json** y en **autoload.php**. El proceso es igual de sencillo:

Únicamente ejecutamos la instrucción `remove composer remove PHPMailer\PHPMailer`



INSTALACIÓN Y ACTUALIZACIÓN DE PAQUETES INCLUIDOS EN COMPOSER.JSON

El proceso de instalación se puede dar si instalamos paquetes que incluimos directamente en composer.json sin utilizar el CLI o en un nuevo equipo de desarrollo donde no tuvimos que copiar todo el contenido de **vendor**.

Únicamente corremos la instrucción `composer install` desde la raíz del proyecto.

```
Windows PowerShell
PS D:\SOFTWARE\Xampp\htdocs\GIT\test> composer install
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing phpmailer/phpmailer (v6.0.7): Loading from cache
phpmailer/phpmailer suggests installing psr/log (For optional PSR-3 debug logging)
phpmailer/phpmailer suggests installing league/oauth2-google (Needed for Google XOAUTH2 authentication)
phpmailer/phpmailer suggests installing hayageek/oauth2-yahoo (Needed for Yahoo XOAUTH2 authentication)
phpmailer/phpmailer suggests installing stevenmaguire/oauth2-microsoft (Needed for Microsoft XOAUTH2 authentication)
phpmailer/phpmailer suggests installing symfony/polyfill-mbstring (To support UTF-8 if the Mbstring PHP extension is not
 enabled (^1.2))
Writing lock file
Generating autoload files
PS D:\SOFTWARE\Xampp\htdocs\GIT\test>
```

El proceso de actualización es similar y nos puede servir para actualizar un paquete en particular o todos los paquetes instalados por composer al proyecto.

Para un paquete, ejecutamos `composer update PHPMailer\PHPMailer`. Para actualizar el proyecto completo, `composer update`

```
PS D:\SOFTWARE\Xampp\htdocs\GIT\test> composer update
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Generating autoload files
PS D:\SOFTWARE\Xampp\htdocs\GIT\test>
```

BOWER PARA JAVASCRIPT ES5

Este manejador está diseñado para librerías javascript que se deseen manejar en la modalidad antigua (ES5) de inclusión de scripts a sitio o app, es decir incluyendo los script tags dentro del código fuente:

```
<script src="lib/jquery/jquery.js"></script>
```

Nota: Aunque perfectamente funcional, veremos más adelante que la metodología de trabajo ES6 nos ofrece ciertas ventajas por sobre ES5.

REQUERIMIENTOS PREVIOS

- Node & NPM
- GIT

INSTALACIÓN

Una vez instalados los requerimientos **bower** se instala directamente sobre el CLI de NPM, con el comando:

```
npm install -g bower
```

Una vez finalizada la instalación podremos utilizar bower mediante su CLI; esto se realiza dentro de la carpeta Public o App del proyecto ya que en este caso las dependencias son instaladas para el frontend.

Nota: Por defecto bower instala las librerías en una carpeta **bower_components**. Para cambiar esto debemos crear previamente en la raíz de public un archivo nombrado **.bowerrc** con el nombre del directorio donde queremos almacenar nuestras dependencias; en nuestro caso utilizamos por estándar la carpeta **lib**.

```
{
    "directory": "lib"
}
```

INSTALACIÓN DE PAQUETES

Todo el proceso se realiza desde el cli de bower con 2 consideraciones:

- El paquete/libreria se encuentra dado de alta en el listado de bower:

```
bower install jquery --save
```

Ya que jquery está listado en los paquetes de bower solo necesitamos su nombre; otros ejemplos serian: fontawesome, bootstrap, angular.

- El paquete/libreria no se encuentra listado, pero hay un github o liga al script:

```
bower install git://github.com/twbs/bootstrap.git --save
```

o bien:

```
bower install http://example.com/script.js --save
```

Si bien boostrap es soportado por nombre eso no impide instalarlo desde su liga de github como se ve en el ejemplo.

Si requerimos una versión específica podemos indicarla en el caso de un paquete Github o del listado de bower:

```
bower install jquery#1.2.2 --save
```

Esto instalaría jquery específicamente en la versión 1.2.2.

Nota: Podemos cambiar el sub parámetro `--save` por `save-dev` para dependencias que solo utilizaremos en desarrollo.

Las instalaciones generan o actualizan un archivo **bower.json** que se crea en la raíz de ejecución, cuya función es la misma que la de **composer.json**, dar seguimiento a los paquetes instalados y dar la facilidad de reinstalarlos o actualizarlos a versiones especificas. Ejemplo de bower.json:

```
{
  "name": "Frontend de prueba",
  "authors": [
    "Gobierno del Estado de Yucatán"
  ],
  "license": "MIT",
  "homepage": "www.yucatan.gob.mx",
  "private": true,
  "ignore": [
    "**/.*",
    "node_modules",
    "bower_components",
    "test",
    "tests",
    "lib"
  ],
  "dependencies": {
    "jquery": "1.2.23",
    "bootstrap": "~3.2.0",
    "animate.css": "~3.2.0",
  }
}
```

Recuerda que a diferencia de composer los paquetes de bower no ofrecen un punto único de integración por ser utilizados bajo metodología ES5, por lo que debes ingresar los scripts correspondientes en los lugares necesarios, para jquery, por ejemplo:

```
<script src="lib/jquery/dist/jquery.min.js"></script>
```

DESINSTALAR PAQUETES

Las consideraciones son las mismas que en install, pero debemos utilizar el comando uninstall:

```
bower unistall jquery
```

ACTUALIZAR PAQUETES

De la misma forma que composer funciona en modalidad general y especifica:

```
bower update jquery --save
bower update
```

Nota: El sub parámetro `--save` es necesario para actualizar el listado en bower.json; de no utilizarlo el paquete no quedará registrado como dependencia de la aplicación, o en su defecto la versión actualizada.

NPM PARA FRONTENDS ES6 O BACKEND NODEJS

No entraremos en detalle amplio de manejo de Apps en formato ES6 ya que es un tema extenso, sin embargo, su uso ofrece diversas ventajas sobre aplicaciones web tradicionales:

- Mejor manejo de paquetes
- Mejor manejabilidad del código
- Uso de librerías o código Javascript avanzado no disponible en navegadores web, pero compatible en compilación
- Uso de variantes de ECMAScript adicionales a javascript (TypeScript o CoffeeScript)
- Modificación y compilación de código automático en distribución
- Eliminación de elementos innecesarios en distribución (imágenes no utilizadas, README, partes de librerías que no se utilizaron, etc.).
- Trabajar frontend y backend con un mismo lenguaje (NodeJS)

Si estás interesado en mejorar tus habilidades y programar frontend con ES6 aquí algunos recursos.

- How to Build a WebPack App with vanilla Js - Utilizando React
- Laracast Vue Step by Step - Utilizando framework vue
- Angular Deployment - Utilizando angular

Basado en estos recursos puedes construir cualquier app en ES6 y aprender de algunos frameworks disponibles.

INSTALACIÓN

Si trabajas un proyecto ES6 te darás cuenta que el manejador NPM trabaja directamente relacionado al desarrollo durante toda la vida del proyecto; únicamente necesitas descargar una distribución de NodeJs para comenzar:

- NodeJs Sitio Oficial

Durante el proceso, el node en Windows te solicitará permiso para instalar el npm e incluirlo en el PATH de sistema. Se deben aceptar estas opciones para poder continuar.

INSTALACIÓN DE PAQUETES

En una estructura de proyecto node todos los paquetes se almacenan en una carpeta **node_modules**; al igual que bower o composer, las dependencias se guardan en un archivo **package.json y package.lock** donde están divididas en:

- dependencies - Para dependencias del proyecto
- devDependencies - Para dependencias de desarrollo únicamente

Entre otros datos de información del proyecto, las instalaciones se realizan mediante el comando:

```
npm install --save jquery
```

ó

```
npm install --save-dev lodash
```

Las instalaciones a versiones específicas de paquetes, requieren la inclusión de una @ y el número de versión:

```
npm install --save lodash@4.17.4
```

Al igual que en bower el parámetro --save indica que el paquete será incluido en el listado de dependencias del **package.json**.

Por último, podemos ejecutar el comando de instalación sin un paquete específico; esto instalará todas las dependencias listadas en package.json en nuestro proyecto:

```
npm install
```

Ejemplo de package.json

```
{
  "name": "gob.mx.yucatan.vue.boilerplate",
  "description": "Boilerplate inicial para aplicaciones Vue",
  "version": "1.0.0",
  "author": "Gobierno del Estado de Yucatan",
  "license": "MIT",
  "private": true,
  "scripts": {
    "dev": "cross-env NODE_ENV=development webpack-dev-server --
open --inline --hot",
    "webpack": "cross-env NODE_ENV=development webpack --watch",
    "build": "cross-env NODE_ENV=production webpack --progress -
-hide-modules"
  },
  "dependencies": {
    "axios": "^0.18.0",
    "popper.js": "^1.14.3",
    "vue": "^2.5.11",
  },
  "browserslist": [
    "> 1%",
    "last 2 versions",
    "not ie <= 8"
  ],
```

```
  "devDependencies": {
    "babel-core": "^6.26.0",
    "babel-loader": "^7.1.2",
    "babel-preset-env": "^1.6.0",
    "babel-preset-stage-3": "^6.24.1",
    "webpack": "^3.6.0",
    "webpack-cli": "^2.0.14",
    "webpack-dev-server": "^2.9.1"
  },
  "main": "webpack.config.js",
  "keywords": []
}
```

DESINSTALAR PAQUETES

La desinstalación es muy similar a bower; únicamente indicamos el paquete:

```
npm uninstall --save lodash
```

En este caso la instrucción save es requerida para indicar que el paquete también será removido del package.json.

ACTUALIZAR PAQUETES

La actualización de paquetes funciona de la misma forma; únicamente indicamos el paquete a actualizar o si no se indica, se realizará una actualización global de paquetes.

```
npm update--save-dev lodash
```

ó

```
npm update
```

CAPACIDADES DE ES6

Los paquetes instalados y trabajados en proyectos ES6 permiten importarse entre las diversas librerías o clases del proyecto de la siguiente forma:

```
import _ from 'lodash'
import $ from 'jquery'
function foo(num) {
  let result = _.sum(num + 5);
  $('#div').html(result);
}
```

No dependemos de que se importe un script tag en el html porque el proyecto base está corriendo sobre una versión avanzada de javascript como su motor principal, y que generalmente compilaremos en una distribución que un navegador pueda interpretar.

Una ventaja más de este modelo de trabajo es que podemos trabajar apps es6 que sean corridas directamente por un servidor NodeJs, lo que nos permitiría en un momento dado eliminar la dualidad de lenguajes y trabajar una aplicación o sitio web únicamente con Javascript ES6, tanto en el frontend, como en el backend (sin necesidad de php, rails, python, etc.).

Para más información sobre cómo publicar servicios basados en NodeJs puedes verificar los siguientes recursos:

- Simple Web Server with NodeJs
- NodeJs Express App Tutorial

Actualmente contamos en el repositorio con el Boilerplate para una app ES6 basada en Vue. Puedes verificar el recurso en la siguiente liga:

- Vue Boilerplate

REPOSITORIOS DE CÓDIGO

Un repositorio de código es un lugar donde el código de una aplicación o de un programa cualquiera está almacenado y desde donde se puede distribuir. Hoy en día debe ser una práctica mandatoria, dada la creciente necesidad de tener un control modular, recuperación, manejo de cambios y trabajo en equipo.

GITLAB

Actualmente utilizamos la herramienta GitLab ya que permite el uso de un repositorio con funciones de trabajo colaborativo y seguimiento de avance de proyectos (mejoras, desarrollos y bugs) y puede ser instalado y administrado de manera privada en nuestros servidores. La liga actual es:

Servidor GitLAB de Gobierno del Estado

De igual forma, tenemos un flujo completo de trabajo propuesto que se basa en el uso y explotación de la herramienta. La documentación completa la puedes encontrar en:

- Presentación en slides
- Documento de la presentación

En la documentación se explica a detalle tanto el flujo de trabajo propuesto como las ventajas y funcionalidades que se pueden aprovechar en las etapas de desarrollo con el GitLab. En este documento se tocarán únicamente los puntos básicos.

- Proyectos
- Grupos
- Usuarios
- Roles de trabajo

PROYECTOS DE GITLAB

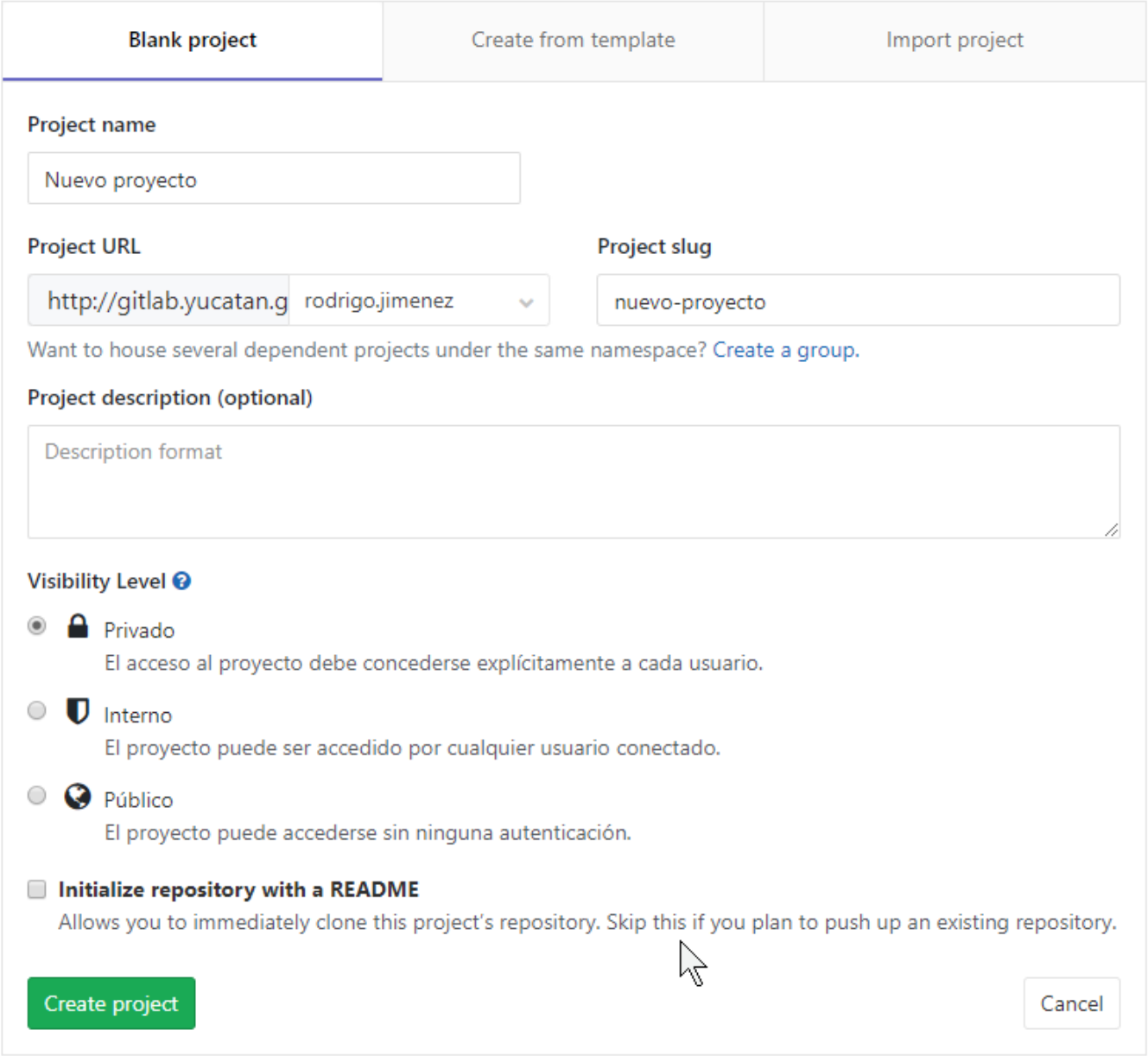
Un proyecto es básicamente un espacio en el repositorio destinado a mantener y controlar el flujo de desarrollo de un sistema/página/plataforma o app en particular.

Un proyecto puede ser:

- Público
- Privado
- Interno

A su vez, un proyecto puede pertenecer a:

- Grupos
- Usuarios



Diferencias

- Un proyecto **privado** requiere acceso explícito a cada usuario para que pueda utilizarlo; esta restricción no aplica si el proyecto pertenece a un **grupo** y ese grupo tiene otros **maintainers**.
- Un proyecto **interno** puede ser accedido por cualquier usuario conectado al Gitlab.
- Un proyecto **público** puede ser accedido por cualquier usuario que tenga acceso al portal de Gitlab, aun cuando no esté registrado; esto no quiere decir que pueda realizar cambios, este modo podría estar restringido si el proyecto se está creando dentro de un grupo privado.

PROYECTOS DE USUARIO O GRUPO

Estas dos vertientes tienen las siguientes diferencias:

- En un proyecto de usuario, el usuario individual tiene mayor control sobre el contenido; puede él mismo, dar accesos específicos a su proyecto tan público o privado como desee.
- Un proyecto de grupo está más restringido en el sentido de que no puede privatizarse para usuarios que pertenezcan al mismo grupo en cuanto a visibilidad.

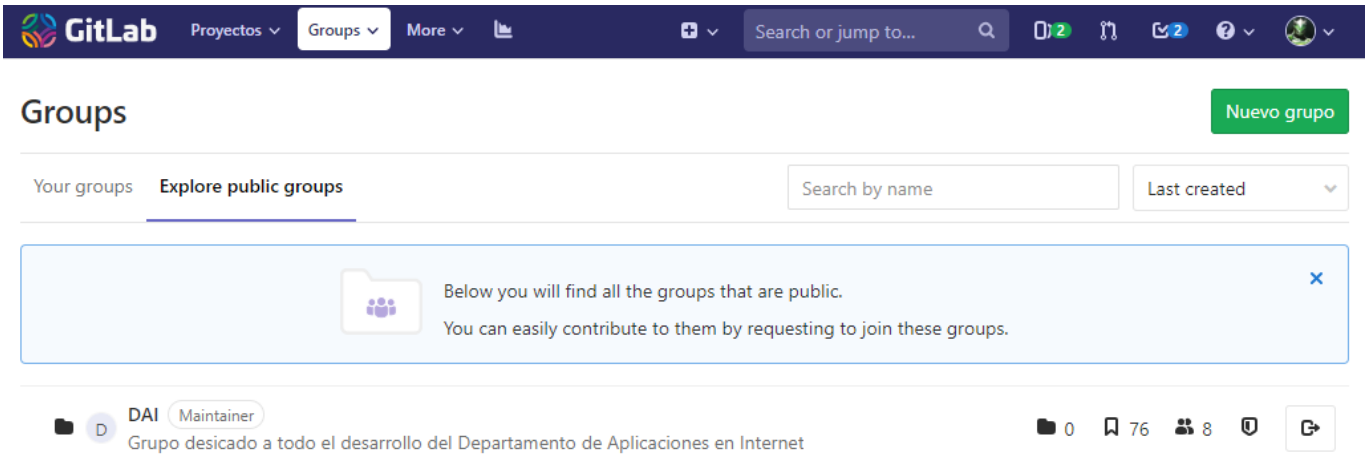
OPCIONES DE CREACIÓN DE PROYECTOS

Un proyecto puede crearse en gitlab mediante 3 vertientes:

- **Proyecto en blanco**. Como su nombre lo indica, crea un repositorio vacío inicialmente.
- **Proyecto desde template**. Estos proyectos están basados en templates públicos de Gitlab o templates privados que la administración de Gitlab de Gobierno ponga a disposición como un template.
- **Importar proyecto**. Esta última opción nos permite importar un proyecto existente a nuestro repositorio privado; la importación puede ser desde otro Gitlab, github, bitbucket o en general cualquier otro repositorio git al que tengamos acceso.

GRUPOS

Los grupos son una forma de organizar a los proyectos y los desarrolladores. Puedes crear grupos en Gitlab para separar los desarrollos de las diversas áreas, y los desarrolladores pueden tener diferentes atribuciones en los proyectos de grupos diferentes.



USUARIOS

Los usuarios se dan de alta por la administración y pueden tener diversas atribuciones en los grupos y proyectos. Se manejan 2 roles principales:

- Developers
- Maintainers

La diferencia principal es que los **Maintainers** tienen algunas atribuciones especiales, pueden proteger ramas de proyectos (como **master**) y dar o negar permisos a los **Developers** en los proyectos que manejan.

ROLES DE TRABAJO

Mencionados anteriormente, tienen la funcionalidad de determinar el flujo del trabajo, Según el tamaño del proyecto, los roles de trabajo pueden tener un impacto diferente en los proyectos. Éstas son algunas de las facultades que se pueden manejar mediante los roles de trabajo:

- Maintainer puede dar acceso a proyectos a developers y programar **webhooks** para captura de eventos.
- Maintainer tiene acceso a **master** y **tags**.
- Developer requiere acceso explícito a proyectos no públicos.
- Developer puede crear ramas de trabajo basadas en **master** pero no modificarlo directamente.
- Developer puede hacer **Merge Request** para solicitar unir su rama a **master**.
- Maintainer puede aceptar, denegar o revertir estos merge request.

- Maintainer puede hacer liberaciones manuales o automatizadas de versiones de código.
- Ambos roles pueden asignar y cambiar las tareas levantadas en el sistema **issues**.

Anexo V
Políticas de bases de datos para terceros

Gobierno del Estado de Yucatán

Secretaría de Administración y Finanzas

Política de Base de Datos para Terceros

Revisión 00

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

Contenido

I. Objetivo 3

II. Alcance 3

III. Política general 3

IV. Políticas específicas 3

1. Entrega de documentos 3
2. Solicitud de implementación de cambios 5
3. Solicitud de respaldos de base de datos 6
4. Solicitud de creación y actualización de permisos de usuarios de Base de Datos 7

V. Sanciones 8

VI. Firma de autorización del documento 9

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

I. Objetivo

Definir los procesos necesarios para la gestión de base de datos de Terceros que laboran para el Gobierno del estado de Yucatán.

II. Alcance

Esta política se define con el fin de proporcionar a Terceros que laboran para el Gobierno del estado de Yucatán los procedimientos establecidos para le entrega de documentación, y administración de bases de datos nuevas y existentes en producción.

III. Política general

En materia de creación o actualización de estructuras y objetos de base de datos es necesario que los Terceros que laboran para el Gobierno del Estado de Yucatán cumplan con las normas establecidas en las PL-DED-AAB-01-R00-POLÍTICAS GENERALES DE BASE DE DATOS descrito en el apartado VII.

IV. Políticas específicas

1. Entrega de documentos

Al momento de concluir con el desarrollo y SET de pruebas de un nuevo sistema, es necesario que los Terceros realicen la entrega de la siguiente documentación de base de datos al área de Administración de Base de Datos de la Secretaria de Administración y Finanzas del Gobierno del Estado de Yucatán:

1. Formato de recepción de base de datos F-DED-AAB-05-R00-FORMATO DE RECEPCIÓN DE BASE DE DATOS.
2. Formato de solicitud de usuarios de Base de Datos F-DED-AAB-03-R00-FORMATO DE SOLICITUD USUARIO BASE DE DATOS.
3. Documento de configuración del motor de base de datos y aplicativos que consumen la información resguardada en la base de datos.
4. Documento de estructura y objetos de la base de datos, el documento también debe de contener lo siguiente:
    a. Indicar las tablas que tendrán un alto nivel de transaccionalidad.
    b. Estimación de crecimiento de la base de datos y de las tablas, indicando el número de transacciones esperadas por día.
5. Diccionario de Datos.
6. Modelo Entidad-Relación.
7. Usuarios contraseñas de los usuarios creados en la base de datos.
8. Documento de SET de pruebas realizados en los ambientes QA.

Si la base de datos ya existe en producción y se requiere de la creación de nuevas tablas o usuarios para un nuevo proyecto o actualización del sistema que consume la información de la base de datos, el proveedor deberá de entregar los siguientes documentos:

1. Formato de solicitud de usuarios F-DED-AAB-03-R00-FORMATO DE SOLICITUD USUARIO BASE DE DATOS, este documento solo se entregará si se requiere de la creación de un nuevo usuario de base de datos.
2. Documento actualizado de estructura y objetos nuevos que serán creados en la base de datos, el documento también debe de contener lo siguiente:
    i. Indicar si las tablas nuevas tendrán un alto nivel de transaccionalidad.

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

ii. Estimación de crecimiento de las tablas nuevas, indicando el número de transacciones esperadas por día.

3. Diccionario de datos actualizado con la información de los nuevos objetos creados u modificados.
4. Modelo Entidad-Relación actualizado.

La documentación deberá ser enviada con 15 días hábiles de anticipación, antes de que el sistema sea liberado en producción, debido a que la documentación deberá ser revisada por el área de Administración de Base de Datos para garantizar que cumpla con lo solicitado en las políticas generales de Base de Datos y la Política de Base de Datos para Terceros , en caso de tener observaciones el área de Administración de Base de Datos enviará las correcciones por correo y el Tercero deberá de realizar las adecuaciones indicadas.

Todas las bases de datos nuevas creadas por Terceros para fines administrativos y operativos de las Dependencias y Entidades que conforman el Gobierno del estado de Yucatán, deberán ser gestionadas por el área de Administración de Base de Datos y deberán cumplir con los apartados estipulados en la política de Base de Datos para Terceros.

2. Solicitud de implementación de cambios

Para la implementación de cambios a bases de datos solicitados por Terceros que laboran para el Gobierno del Estado de Yucatán, se deberá de cumplir con los siguientes lineamientos.

1. Comunicar anticipación al área de Administración de Base de Datos sobre la implementación de nuevo cambio.
2. Se deberá contar con los Vo. Bo de las direcciones involucradas en el cambio.
3. Proporcionar el cronograma de actividades del cambio.

4. Entregar la documentación indicada en el apartado uno de este documento.

En caso de no cumplir con los lineamientos establecidos en este apartado, el Tercero será consignado a las autoridades correspondientes.

3. Solicitud de respaldos de base de datos

Para los Terceros que laboral para el Gobierno del Estado de Yucatán y requieran tener acceso a los respaldos de bases de datos por virtud de contrato, para la ejecución de pruebas de los nuevos cambios realizados a la base de datos solicitados por el Gobierno de Estado de Yucatán, deberán cumplir con lo siguiente:

1. Proporcionar el formato de solicitud de respaldo FF-DED-AAB-09-R00-FORMATO DE SOLICITUD DE RESPALDO DE BASE DE DATOS, el cual debe de estar llenado de manera correcta.
2. Deberá de contar con el Vo.Bo. del director de la Secretaría de Administración y Finanzas descrita en correo.
3. El formato de solicitud de respaldo F-DED-AAB-09-R00-FORMATO DE SOLICITUD DE RESPALDO DE BASE DE DATOS deberá ser enviado al área de Administración de Base de datos con copia a la dirección de Tecnologías de la Información y con el Visto Bueno adjunto de la dirección de Secretaria de Administración y Finanzas.

El respaldo será entregado en los tiempos establecidos en las políticas generales de Administración de Base de Datos.

La información será entregada de la siguiente forma:

1. Toda información sensible de base de datos deberá ir enmascarada, se entiende por información sensible:
    a. Nombre y apellido de usuarios

b. Número telefónico
c. Preferencia sexual
d. Estado de salud
e. Creencias religiosas.
f. Afiliación sindical

2. El respaldo será creado con el formato correspondiente según sea el Sistema de Gestión de Base de datos (RDBMS, por sus siglas en inglés).
3. El Tercero deberá proporcionar una conexión segura para la transferencia del respaldo, haciendo uso del protocolo SFTP.
4. Una vez concluido las mejoras y pruebas para el cual fue solicitado el respaldo, el Tercero deberá de garantizar el borrado del respaldo proporcionado.

4. Solicitud de creación y actualización de permisos de usuarios de Base de Datos

Para los Terceros que prestan sus servicios al Gobierno del Estado de Yucatán, y requieran de realizar la solicitud de creación de usuarios de Bases de datos por concepto de mejoras o creación de nuevos módulos en un sistema, el líder del cambio deberá de entregar la siguiente documentación:

1. Formato de Solicitud de Usuarios de Base de Datos F-DED-AAB-03-R00-FORMATO DE SOLICITUD DE USUARIODE BASE DE DATOS llenado de manera correcta.
2. Deberá de contar con el Vo.Bo. del director de la Secretaria de Administración y Finanzas descrita en correo.
3. El formato de Solicitud de Usuarios de Base de datos deberá ser enviado por correo al área de Administración de Base de Datos con el correo adjunto del Vo.Bo. del director de la Secretaria de Administración y Finanzas.

Concluida las pruebas funcionales y de estrés en los ambientes de QA-Testing, al realizar su pase a producción el Tercero deberá de hacer entrega del usuario y contraseña al área de Administración de Base de Datos para el cambio de contraseña y resguardo del usuario.

Para los terceros que requieran de la actualización de permisos a usuarios existentes en la base de datos, el líder del cambio deberá de entregar la siguiente documentación:

1. Formato actualizado de Solicitud de Usuarios de Base de Datos F-DED-AAB-03-R00-FORMATO DE SOLICITUD DE USUARIO BASE DE DATOS, indicando los nuevos permisos u objetos de base de datos a los cuales requiere que el usuario tenga acceso.
2. Vo.Bo. del director de la Secretaria de Administración y Finanzas descrita en correo.
3. El formato deberá ser enviado por correo al área de Administración de Base de Datos justificando el motivo por el cual son requeridos la actualización de permiso u objetos de Base de datos con el correo adjunto del Vo.Bo. del director de la Secretaria de Administración y Finanzas.

V. Sanciones

Cualquier apartado indicado en este documento que no sea cumplido por los Terceros que laboran para el Gobierno del Estado de Yucatán, serán consignados con las autoridades correspondientes para que se apliquen las sanciones establecidas.

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

VI. Firma de autorización del documento

Autorizó

_____

Nombre

Puesto el titular

[Fin del documento]

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

Anexo VI
Políticas para bases de datos geoespaciales

Gobierno del Estado de Yucatán

Secretaría de Administración y Finanzas

POLÍTICA PARA BASES DE DATOS GEOESPACIAL

Revisión 01

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

Índice

I. OBJETIVO 3

II. ALCANCE 3

III. FUNDAMENTO LEGAL 3

1. Ámbito Federal 3

2. Ámbito Estatal 3

IV. DEFINICIONES 4

V. POLÍTICAS 7

1. Políticas Generales 7

1.1. Política de transferencia de información 7

1.1. Política de almacenamiento y manejo de datos 8

1.2. Política de representación de la información 8

2. Políticas Específicas 8

2.1. Política de Transferencia de Información 8

3. Política de almacenamiento y manejo de datos 12

4. Política de representación de la información 14

VI. FIRMA DE AUTORIZACIÓN DEL DOCUMENTO 15

I. OBJETIVO

Para la protección de datos y el tratamiento de éstos como objetivo general de la STIC está el asegurar la confidencialidad, integridad, libertad, veracidad, transparencia y disponibilidad de la información y bases de datos de las Dependencias y Entidades de la Administración Pública Estatal, garantizando disponibilidad de la infraestructura tecnológica para el tratamiento de los datos recolectados para fines legales, contractuales, y comerciales.

Buscando siempre mecanismos efectivos para la protección de los derechos de los titulares de los datos a los cuales les dé Tratamiento y, gestionará y desarrollará medidas que contengan condiciones de seguridad adecuadas para evitar la adulteración, pérdida, consulta, uso o acceso fraudulento sobre la información.

II. ALCANCE

Aplica a las Dependencias y Entidades de la Administración Pública Estatal.

III. FUNDAMENTO LEGAL

1. Ámbito Federal

No aplica.

2. Ámbito Estatal

Artículo 66. Fracción V; del reglamento del código de la administración pública.

IV. DEFINICIONES

Sistema de coordenadas cartesianas: sistema de coordenadas que da la posición de puntos respecto de n ejes mutuamente perpendiculares.

Sistema de referencia de coordenadas sistema de coordenadas que está referido al mundo real a través de un datum.

Datum: parámetro o conjunto de parámetros que sirven como referencia o base para el cálculo de otros parámetros.

Sistema de coordenadas: conjunto de reglas matemáticas que especifican cómo las coordenadas deben ser asignadas a los puntos.

Coordenada: cualquiera de los n números de una secuencia que designa la posición de un punto en un sistema n-dimensional.

Conversión de coordenadas: los sistemas de origen y destino comparten el mismo datum. Es una transformación exacta y se basa en la aplicación de fórmulas establecidas que relacionan ambos sistemas;

Transformación de coordenadas: El datum es distinto en los sistemas de origen y destino.

European Petroleum Survey Group (EPSG): es el consorcio petrolífero que elabora un esquema de modificación específico con información acerca de los distintos sistemas de coordenadas de fácil acceso y manejo, cada sistema de coordenadas se relaciona con un código EPSG.

Proyección cartográfica: conversión de coordenadas desde un sistema de coordenadas geodésicas a uno plano.

Altitud: distancia a un punto desde una superficie de referencia elegida a lo largo de una normal a esa superficie.

Punto: un objeto cero-dimensional que especifica la localización geométrica por medio de un conjunto de coordenadas.

Línea: un objeto unidimensional que es un segmento de línea determinado entre dos puntos.

Polígonos: El área es dividida en un conjunto de elementos poligonales, de tal forma que cualquier localización que cae dentro de un polígono tiene asociado un valor y las fronteras son definidas por un conjunto de parejas ordenadas (x, y) de puntos.

Localización y extensión: Es la ubicación y alcance de las coordenadas (x, y, z) en el sistema de referencia específico. Estos aspectos están representados por cualquiera de los puntos, líneas y polígonos.

Valores temáticos: Un objeto tiene varios atributos que definen sus valores.

Calidad de la información: La calidad describe el valor de los datos. Esta información es importante al evaluar la credibilidad de los datos.

Error: Discrepancia existente entre el valor real (puede ser un valor de precisión, de un atributo o cualquier otro) y el valor recogido en una capa.

Precisión: Nivel de detalle con el que se recoge la información.

Exactitud: Indica el grado en que los valores estimados se asemejan al valor real.

Exactitud posicional: La referencia geográfica que tiene el dato espacial, la precisión con la que se toma condiciona la calidad del dato.

Exactitud de los atributos: Los valores asociados a una coordenada u objeto espacial pueden haber sido medidos con más o menos exactitud, o presentar valores incorrectos.

Coherencia lógica y coherencia topológica: atributos y valores coherentes con escalas de medida o el tipo de valor que se espera.

Compleción: Elementos pueden no haberse recogido por cuestiones de escala (menores del tamaño mínimo), pero también pueden incluirse o excluirse en función de otros criterios.

Calidad temporal: La realidad que representa un dato geográfico en una realidad que varía con el paso del tiempo, y por tanto se puede degradar la calidad del dato espacial en mayor o menor medida.

Procedencia: fuente o fuentes de donde se obtiene la información y generación, número de procesos que pudieran generar algún tipo de error.

Incertidumbre: Es cuando el valor real es desconocido por lo que el error no puede conocerse, la incertidumbre releja la medida que no podemos tener certeza de validez de nuestros datos.

Vaguedad: Aparece como consecuencia de definiciones pobres e incompletas, así como cuando los objetos que se modelizan en los datos no presentan límites bien definidos.

Ambigüedad: Cuando no hay definiciones inequívocas de los conceptos fundacionales aparecen ambigüedades que añaden incertidumbre al dato creado en función de estos.

Escala: Relación entre el tamaño real de aquello que representamos y su tamaño en la representación básico de toda información geográfica.

Escala cartográfica: Relación entre el mapa y lo real.

Escala de análisis u operacional: Define la utilidad de los datos y lo que podemos hacer con ellos, ya que indica sus limitaciones.

Dato: simple conjunto de valores que se utiliza para representar algo.

Significado del dato: Es el que proporciona información.

Datos primarios: GPS o alguna herramienta especializada para su captura.

GPS o Sistema de Posicionamiento Global (Global Positioning System): es un sistema de navegación basado en satélites.

Datos secundarios: Derivan de algún dato previo que hay que tratar.

Teledetección: Estudio y medida de las características de una serie de objetos sin que exista un contacto físico; se miden las perturbaciones que el objeto provoca en su entorno.

Errores en la digitalización: registro inexacto, puntos mal situados, errores de registro, desplazamiento por vértices insuficientes.

Estándar: la especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad.

Conjunto de Datos Espaciales: la totalidad de los datos que corresponden a un área geográfica con límites y escala determinados.

Atributo: la propiedad de los objetos, que describe sus características geométricas, topológicas u otras;

Metadatos: conjunto de elementos que describen las características de los datos y servicios, que permiten a un productor de información geográfica describir los recursos geográficos de modo que los usuarios puedan conocer sus características y puedan evaluar la aplicabilidad de los recursos para un uso determinado.

SIG: Sistema de Información Geográfica.

FGDC: Content Standard for Digital Geoespatial Metadata.

V. POLÍTICAS

1. Políticas Generales

1.1. Política de transferencia de información

Establece criterios de estandarización para la transferencia de datos que conserven la calidad de la información siguiendo un proceso adecuado desde su captura hasta su entrega con el propósito de que estos cumplan su función.

1.1. Política de almacenamiento y manejo de datos

Establece criterios de estandarización y homologación para la integración de datos espaciales a una base de datos, en la cual, estos se almacenan de manera homogénea estructuralmente y se asegura la compatibilidad de la información, generando con ello procesos de vinculación e integración más ágiles y confiables.

1.2. Política de representación de la información

Establece criterios para la representación de información espacial visualizada por medio un Sistema de Información Geográfica (SIG), a partir de objetos integrados a una base de datos.

2. Políticas Específicas

2.1. Política de Transferencia de Información

La información que es generada por cada institución de nivel estatal debe ser proporcionada de acuerdo con lo establecido en el artículo 72 de la Ley de Transparencia y Acceso a la Información Pública del Estado de Yucatán y el Artículo 70 de la Ley General de Transparencia y Acceso a la Información Pública.

En relación con el proceso interno de obtención de datos se establecen 2 casos, los cuáles deben seguir diferentes criterios de transferencia:

Si se trata de datos primarios se debe de especificar las características técnicas del instrumento de captura utilizado (GPS, instrumentos del tipo taquímetro o estación total, drones), el sistema de referencia geográfica que arroja el instrumento, así como también la precisión y exactitud a la que proporciona la información.

Nombre técnico del instrumento
Descripción
Periodo de Actualización

Características
Precisión
Exactitud
Software de procesamiento de datos
Archivo de salida

En el caso en el que sean datos secundarios, se debe describir la fuente de la que provienen, el año de obtención y de actualización, la especificación de cambios en el caso de tener modificaciones de datos o de estructura mencionando el programa utilizado en este proceso.

Fuente
Año_Captura
Año_Actualización
Periodicidad_Actualización
Modificaciones
Programa

El manejo de los datos espaciales se debe realizar a través de archivos compatibles con el Sistema de Información Geográfica;

a. Shapefile (.shp): Este tipo de archivo acompaña un archivo de extensión .shx (índice de entidades geométricas), un archivo con extensión .dbf (base de datos que contiene su información) y un archivo con extensión .pjr (sistema de proyección).
b. Csv (delimitado por comas): Compatible con Excel y que debe contener los campos con latitud y longitud de acuerdo con el sistema de coordenadas establecido este documento.

c. Keyhole Markup Languaje (Kml o Kmz): Es el de más fácil acceso y edición, pero que no es recomendado si se requiere precisión en la información.

Se debe de evitar problemas de digitalización de la información al ingresarla con algún programa relacionado con captura de datos espaciales; lo más recomendado es utilizar instrumentos especializados que otorguen mayor precisión.

El sistema de coordenadas o sistema de proyección a utilizar es el siguiente:

Estas coordenadas geográficas deben estar al Dátum WGS84 en este los valores de la latitud y longitud se representan en grados decimales anteponiendo el signo negativo (-) para la longitud y con al menos siete decimales (Anexo 1: Tipos de datos de PostgreSQL);

Cónica Conforme de Lambert
Esferoide.....WGS84
Proyección.....Cónica Conforme de Lambert
1er Paralelo base.....17° 30′ 00″
2o Paralelo base..... 29° 30′ 00″
Meridiano central.....–102 00′ 00″
Latitud de origen de la proyección.....12° 00′ 00″
Falso este (metros).....2500 000
Falso norte (metros).....0.00
Datum Horizontal.....WGS84
EPSG.....4326

El archivo debe estar definido por:

Un tipo de geometría; Punto (X, Y) polígono (X, Y por vértice, rumbo, distancia y superficie, su característica principal es que está formado por una línea cerrada completamente) o línea (X, Y por vértice, rumbo, distancia) que deben de estar correctamente configurados de acuerdo con sus características particulares.

LADO		RUMBO	DISTANCIA	V	COORDENADAS	
EST	PV				Y	X
				1	2,360,633.0000	253,670.0000
1	B	N 81°41'06.55" E	12.500	B	2,360,634.8077	253,682.3686
B	C	N 01°12'35.08" E	162.960	C	2,360,797.7313	253,685.8091
C	D	S 84°09'48.13" W	12.500	D	2,360,796.4602	253,673.3739
D	1	S 01°10'56.80" W	163.495	1	2,360,633.0000	253,670.0000
SUPERFICIE = 2,018.645 m2						

Identificador (id); Números enteros consecutivos, ninguno debe de repetirse, no debe contener ningún NULL.
Columnas: Deben estar en minúsculas, sin espacios (los cuales pueden ser sustituidos por guion bajo [_], la información debe estar codificada en UTF-8 para ser procesada de manera adecuada sin caracteres difíciles de interpretar por el sistema (ñ, #, ",?,¡,´etc...).

Si la información se encuentra en formato CSV (delimitado por comas), debe de contener los campos en donde se indique la posición de los datos dependiendo del tipo de geometría del que se trate y siguiendo el sistema de referencia.

Para la correcta utilización de la información en procesos posteriores se debe presentar de manera detallada la caracterización de los datos que se van a entregar (Anexo 2: Metadatos)

Si el documento de salida de la información es de tipo CAD, este debe estar de acuerdo con los estándares de referencia espacial que establece este documento (escala y sistema de referencia adecuada), en el caso de no presentar estas características se proporcionará un formato al cual el usuario tendrá que adaptar la

información para su correcta utilización poniendo la información relacionada con las capas que se pretenden anexar; si estas requieren atributos se tendrá que vincular con un Excel en el que se detalle lo que contendrá y posterior a esto se debe realizar una verificación del resultado.

En el caso de que una capa no tenga la proyección adecuada se debe generar la re-proyección y caracterizar los cambios que tuvieron los datos.

3. Política de almacenamiento y manejo de datos

El formato adecuado para integrar la información a la base de datos es de tipo shapefile; el archivo y su contenido deberá cumplir con las características establecidas en la política de transferencia de información (sistema de referencia geográfica, columnas base y metadatos).

La integración de los datos a la base de datos se debe realizar por medio de un manejador de base de datos; este es Postgres 9.6 o superior y PostGIS que le agrega la característica de uso de objetos espaciales; de igual forma se puede utilizar otros tipos de manejadores espaciales, incluyendo aplicaciones tipo GIS (Sistema de información geográfica), MMS (Sistema de Información Multimedia)o CAD (Diseño asistido por Computadora) que sirve para capturar, almacenar, manipular analizar, gestionar y representar espacios geográficos;

- ArcMap (ArcGIS)
- Oracle gis
- ArcInfo (Esri)
- MGE
- TIGRIS (Intergraph)
- Quantum GIS
- GeoMedia

Los criterios base para la conformación de la base de datos son los siguientes:

- Elementos espaciales

- Modelos de datos
- Metadatos
- Sistemas de metadatos

El modelo conceptual de los objetos debe estar formado por tres tablas: geometry_colum, feature_table y spatial_ref_sys, los cuáles hacen referencia a la geometría del objeto, a su estructura y al sistema de referencia geográfica.

Al hacer la transferencia de la información a la base de datos el archivo shapefile se convierte en una tabla y esta deberá de tener las siguientes características:

a. Nombre_ En minúsculas, sin espacios (los cuales pueden ser sustituidos por guion bajo [_] y sin caracteres difíciles de interpretar por el sistema (ñ, #,", ?, ¡,´etc...).
b. Columnas_ geometría (the_geom) que es un campo en formato binario y un identificador (id); ambos con las características establecidas en la *Política de transferencia de información* y se debe especificar el tipo de datos por columna en los Metadatos (Anexo 2) según lo establecido en el Anexo 1: Tipos de datos de PostgreSQL.
c. No debe de contener más de un campo identificador y de geometría y en el caso de que los tenga especificar en los Metadatos (Anexo 2) cuales son los que se utilizan.
d. Los nombres de las columnas deben estar en minúscula y sin espacios.

Todo el contenido de la tabla debe estar codificado en UTF-8 para ser procesada de manera adecuada y sin errores.

En los Metadatos (Anexo 2) se debe especificar si la tabla es de tipo estática o dinámica, así como los nombres de los campos que sufren cambios, la periodicidad de modificación y actualización y la finalidad de este proceso.

La diferenciación de los tipos de tablas es requerida ya que los datos dinámicos deben estar separados de los estáticos por medio de accesos diferenciados, de igual forma se puede presentar más de una tabla dinámica por una de geometría.

4. Política de representación de la información

Para una adecuada visualización de la información integrada a la base de datos y que se representará en un Sistema de Información Geográfica se debe tomar en cuenta su objetivo y las características que se requieren resaltar para su uso, por esto se deberán definir algunas características (Anexo 2: Metadatos).

El nombre de la tabla tal y como aparece en la base de datos.

Para generar la visualización adecuada de la información se debe definir el objetivo de su uso, la precisión requerida para que cumpla con su función, el margen de error en la representación (en el caso de tener modificaciones en el proceso de adecuación o actualización), entre otros puntos; estos relacionados con el objetivo descrito.

Se debe especificar el tipo de uso que se le dará a la información para establecer los accesos y el nivel de interacción.

Generar un esquema de simbología requerida; esto es una descripción detallada de la categorización de la información, colores y tipo de símbolo a utilizar por nivel, así como los nombres que se asignen a cada uno (es lo que se visualiza en la simbología).

La información por utilizar para la visualización será designada por los usuarios; describiendo el nombre de la columna y los atributos designados para cada nivel en la simbología.

En el caso de que se utilice la misma simbología para diferentes tablas, estas deben tener nombres de campo y categorización idénticos y si una contiene más número de niveles que la otra, estos deberán estar homologados.

VI. FIRMA DE AUTORIZACIÓN DEL DOCUMENTO

Autorizó

_____

Nombre

Puesto el titular

[Fin del documento]

Anexo VII
Procedimiento de entrega de documentos de base de datos para terceros

Gobierno del Estado de Yucatán

Secretaría de Administración y Finanzas

Procedimiento de Entrega de Documentos de Base de Datos para Terceros

Revisión 00

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

Contenido

I. Objetivo 3

II. Alcance 3

III. Referencias 3

IV. Definiciones: 4

V. Responsabilidades y autoridad 4

VI. Desarrollo 6

VII. Firma de autorización del documento 7

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

I. Objetivo

Establecer los lineamientos para la entrega de documentos de los proyectos creados e implementados por terceros a la Subsecretaría de Tecnologías de Información y Telecomunicaciones del estado de Yucatán, en relación al motor de base de datos Progress.

II. Alcance

El alcance de este documento comprende el proceso de entrega de documentos de los proyectos efectuados por terceros sobre el DBMS Progress, en las diferentes Entidades y Departamentos del Estado de Yucatán, que están bajo la administración del área de Administración de Base de Datos y Estrategia de Datos de la Dirección de Tecnologías de la Información del Estado de Yucatán.

III. Referencias

PL-DED-AAB-02-R00-POLÍTICA DE BASE DE DATOS PARA TERCEROS.

PL-DED-AAB-07-R00-POLÍTICA DE RESPALDO DE BASE DE DATOS.

F-DED-AAB-03-R00-FORMATO DE SOLICITUD ACCESO A BASE DE DATOS.

F-DED-AAB-06-R00-FORMATO DE RECEPCION DE BASE DE DATOS.

IV. Definiciones:

DBMS: Sistema de Gestión de Base de Datos.

Terceros: Se les conoce como terceros a personas que no labora directamente para una entidad o empresa, sino que prestan sus servicios y se les paga por honorarios, sin goce de prestaciones, también son conocidos como proveedores.

Servidor Storage: Servidor dedicado de almacenamiento, repositorio de ficheros.

V. Responsabilidades y autoridad

Los terceros que laboren para la Subsecretaría de Tecnologías de Información y Telecomunicaciones del Estado de Yucatán que implemente sistemas nuevos en los cuales involucre una Base de Datos nueva sobre el DBMS Progress, deberán de entregar toda la documentación estipulada en la política PL-DED-AAB-02-R00-POLÍTICA DE BASE DE DATOS PARA TERCEROS, cumpliendo con las siguientes responsabilidades determinadas en este documento.

Terceros(proveedores):

Entregar en tiempo y forma los documentos estipulados en las políticas PL-DED-AAB-02-R00-POLÍTICA DE BASE DE DATOS PARA TERCEROS.

Es responsabilidad del proveedor entregar de manera impresa y por correo los documentos solicitados en el primer punto.

Si se realizan cambios a la estructura de la base de datos después de haber entregado el proyecto, como una segunda fase o como mejora, deberá de entregar actualizada la documentación que involucre el cambio.

Administrador de Base de Datos:

Revisar que todos los documentos estipulados en la política PL-DED-AAB-02-R00-POLÍTICA DE BASE DE DATOS PARA TERCEROS sean entregados.

Revisar que los formatos de entrega estén llenados correctamente.

Revisar que el manual de implementación, diccionario de datos, modelo E-R estén bien documentados y completos.

Notificar al proveedor por correo en caso de que algún documento o formato haga falta.

Dar el Vo.Bo. de entrega y cierre del proyecto solo cuando toda la documentación solicitada se entregue de manera correcta y completa.

Si se realizan cambios a la estructura de base de datos como segunda fase o como mejora, el DBA deberá de solicitar al proveedor la actualización de los documentos que involucran al cambio.

Subir en la NAS toda la documentación del proyecto y las actualizaciones de los mismos que con el paso del tiempo se vaya realizando.

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

VI. Desarrollo

Área o puesto responsable	No. Actividad	Descripción
Tercero (Proveedor)	1	Enviar al área de Administración de Base de Datos y Estrategia de Datos la documentación completa del proyecto a entregar.
Administrador de base de datos	2	Validar que la documentación entregada este completa.
Administrador de base de datos	3	¿La documentación entregada está completa? En caso de que sí, continuar con la actividad número 4. En caso de que No, ir a la actividad número 6.
Administrador de base de datos	4	Revisar que la documentación este llenada de manera correcta.
Administrador de base de datos	5	¿Los documentos entregados cuentan con información precisa y correcta? En caso de que sí, ir a la actividad número 10. En caso de que No, ir a la actividad número 7
Administrador de base de datos	6	Notificar sobre los documentos faltantes al proveedor e indicar la fecha en que se requiere sean entregados.
Administrador de base de datos	7	Indicar por correo al proveedor que documento no cuenta con la información completa y solicitar llenarlo de manera correcta o incluir la información faltante.

Tercero(proveedor)	8	Llenar los documentos con la información correcta indicada por el DBA.
Tercero(proveedor)	9	Enviar por correo los documentos actualizados.
Administrador de base de datos	10	Agendar reunión con el proveedor para la firma del acta de entrega de documentos.
Tercero(proveedor)	11	Firmar acta de entrega de documentos.
Administrador de base de datos	12	Firmar acta de entrega de documentos y dar Vo.Bo.

VII. Firma de autorización del documento

Autorizó

_____

Nombre

Puesto el titular

[Fin del documento]

Subsecretaría de Tecnologías de la Información y Comunicaciones
Edificio Administrativo Siglo XXI.
Calle 20-A 284-B por 3-C Colonia
Xcumpich, C.P. 97204 Mérida, Yuc.

T +52 (999) 400 0290
E 42000
yucatan.gob.mx

Anexo VIII

Recomendaciones de seguridad web para considerar en el desarrollo de la página

PRESENTE,

Acciones a considerar para la correcta liberación de un sitio web.

Por lo general, los sitios web, una vez liberados, quedan disponibles para cualquier usuario en la vasta red internacional de internet, y muchas veces, son puestos a prueba malintencionadamente para ganar acceso a recursos, espacio de disco, secuestro de información o de recursos, la mayoría de estos ataques no buscan dejar evidencia de su intención por lo que los atacantes experimentados pueden hacer uso del sistema vulnerado por mucho tiempo antes de que sea descubierto alguna anomalía.

1. Control de acceso a usuarios.

Passwords.
Las contraseñas de acceso pueden encontrarse en diferentes instancias en el proceso de hospedaje de un sitio web, transversalmente, los diferentes dueños de procesos deben tener en cuenta las siguientes consideraciones, y sea en las contraseñas de acceso a la plataforma, al sistema operativo, el usuario con el que corre el servicio, el usuario para la conexión de la base de datos o de la api, etc., deben de ser contraseñas fuertes, en el entendido de que una contraseña fuerte puede componerse de los siguientes componentes:

- Longitud mínima requerida. - Lo habitual es al menos 8 caracteres para evitar descubrimiento rápido por permutación de caracteres.
- Mayúsculas. – Duplica el set de caracteres a evaluar, si se utiliza en combinación con minúsculas.
- Minúsculas. – Duplica el set de caracteres a evaluar, si se utiliza en combinación con mayúsculas.
- Caracteres Especiales, incluyendo espacios en blanco. – La intención es incrementar la aleatoriedad de la posición de un elemento adicional y un juego de caracteres ampliado.
- Números. - Incrementa el set de caracteres a evaluar, si se utiliza en combinación con mayúsculas o minúsculas.
- Complejidad. – Probablemente la más difícil de considerar, ya que por costumbre utilizamos técnicas para recordar las contraseñas, la aleatoriedad puede incrementar el tiempo de convergencia de algoritmos de fuerza bruta, sin embargo, si es suficientemente corta no tiene mucha ventaja contra algoritmos secuenciales de fuerza bruta.

Frase desafío. (Passphrase).
La frase desafío se utiliza comúnmente como una contraseña fuerte, consiste en aprovechar la habilidad nemónica del ser humano para poder referenciar una frase semi encriptada, como ejemplo podemos decir que una contraseña fuerte de acuerdo a los componentes descritos puede ser:

Contraseña	Complejidad	Romper Encriptación

"Gobierno0"	Buena	42 minutos
"Gobierno.0"	Fuerte	10 años
"G0b13rn0.0"	Muy fuerte	10 años
"Gobierno cero.0"	Muy fuerte	34 billones de años
"G0b13rn0 Un0. C3ro"	Muy fuerte	64 billones de años

En el ejemplo, el tiempo que llevaría romper la encriptación hipotéticamente sería utilizando un algoritmo mono proceso serial, las técnicas actuales pueden hacer que equipos poderosos en recursos disminuyan este tiempo estimado. No obstante, queda claro en la comparación, el beneficio de la complejidad de una frase desafío contra una contraseña tradicional, aunado a la facilidad de recordar la frase.

Contraseñas de diccionario.

Existen aplicaciones que se dedican a probar sistemas de autenticación enviando por script las instrucciones de un intento de acceso, y apoyados en bases de datos de contraseñas comunes, conocidas y combinaciones de las mismas. Programas malintencionados (malware) contribuyen a la alimentación y enriquecimiento de estas bases de datos tomando datos del usuario como el nombre o fechas que existan en documentos y agregándolas al diccionario, algunas muestras de diccionario son accesibles desde internet, una buena práctica consiste en no utilizar contraseñas que incluyan palabras que existan en alguna muestra de diccionario, o que asemejen un patrón establecido por estas palabras diccionario. Algunos de los ejemplos más comunes de contraseñas diccionario (idioma inglés):

123456	111111	princess	football	Charlie
password	1234567	admin	123123	aa123456
123456789	sunshine	welcome	monkey	Donald
12345678	Qwerty	666666	654321	password1
12345	Iloveyou	abc123	!@#$%^&*	qwerty123

Compartición de contraseñas.

Salvo sea un requisito de diseño, se desaconseja que más de un usuario utilice las mismas credenciales para cualquier etapa del flujo de la construcción y funcionamiento de una plataforma web. Las contraseñas de sistema o globales deben estar documentadas y debidamente resguardadas pues comprometen la continuidad del servicio de la plataforma web.

2. Software actualizado.

Las plataformas de sistema operativo, servidor web, interprete cgi, apis de desarrollo, paquetes de compilación, librerías de uso público, frameworks de diseño y de scripting, etc, son elementos que comúnmente se encuentran en la composición de una plataforma web.
Las actualizaciones de cada uno de éstos componentes, no solamente incluyen mejoras o adiciones a las características funcionales, sino que también consideran las correcciones de algunos bugs de seguridad, los cuales casi siempre se encuentran documentados. Un atacante puede utilizar la información disponible para una vulnerabilidad específica existente en una versión desactualizada para ganar acceso y privilegios a través de la falla. Contar con un esquema de actualización de paquetes y parches es necesario para conservar una pequeña ventaja en la posibilidad de tener vulnerabilidades bien conocidas y bien documentadas.

3. Registros de actividades y su resguardo.

Los archivos de registro de actividades pueden ser de lo más variado, desde el sistema operativo, intérprete, e incluso en el desarrollo del mismo código de la plataforma pueden existir elementos que constantemente registran en archivos de texto las actividades de cada componente. Éstos elementos de registro son conocidos comúnmente con el nombre de ***Logs***. Se debe contar con un procedimiento que garantice la integridad de estos registros, incluso el respaldo remoto de los mismos. Esto permite trazabilidad y evidencia en caso de algún incidente.

4. Validación básica de paso de parámetros y formularios.

Paso de parámetros
Si es requerido el paso de parámetros, se prefiere el paso de parámetros sensibles mediante el método POST, respetando los requerimientos y mejores prácticas del área de desarrollo involucrado, pues presenta ligeramente mejores beneficios en el tema de seguridad ya que uno pasa los parámetros y sus valores en la url (GET) y el otro lo hace dentro del cuerpo del código (POST), como referencia, presentamos a continuación algunas diferencias clave entre los métodos GET y POST.

Acción	GET	POST
Botón Retroceder / Recargar	Sin Efecto	Los datos serán reenviados (el navegador deberá advertir al usuario de esta situación)
Añadir a Favoritos	Se permite, la url y parámetros quedarán como un acceso directo.	No se pueden agregar los parámetros a una url de favoritos.
Cache	Los datos se almacenan en cache.	No se almacena en cache.
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded o multipart/form-data.

Acción	GET	POST
		(Utiliza multipart encoding para datos binarios)
Historial	Los parámetros quedan almacenados en el historial del navegador.	Los parámetros no son almacenados en el historial del navegador.
Restricciones en longitud de datos.	Si, cuando se envían datos, el método GET agrega los datos a la URL; y la longitud de la URL es limitada (2048 caracteres, y muchos menos caracteres bajo esquemas de protección de equipos firewall).	No hay restricciones.
Restricciones en tipos de datos	Solamente caracteres ASCII son admitidos	No hay restricciones, se admiten datos binarios.
Seguridad	El método GET es menos seguro, comparado con el método POST porque los datos enviados son parte de la URL.	El método POST es un poco más seguro que el método GET, debido a que los parámetros no son almacenados en el navegador, historial, ni en los logs del servidor.
Visibilidad.	Los datos de los parámetros son visibles para todos, por estar en la URL.	Los datos no son visibles en la URL

Validación de controles input.
Adicionalmente, los parámetros pueden estar accionados por el código de la plataforma, como procedimiento interno, un aspecto importante es que los valores de algunos parámetros requieren la entrada del usuario, ésta entrada de datos se realiza por controles input de formularios web.
Los controles input de formularios web, son susceptibles a ser descuidados en las etapas finales de validación al liberar un sitio, un input sin la validación de las cadenas que se almacenan en el valor, puede pasar parámetros directamente al siguiente motor de evaluación, dentro de los cuales, los más comunes son los siguientes:

- SQL Injection.
- XSS attacks.

La mejor manera de mitigar estos riesgos es contar con un procedimiento para validar, escapar y moldear los valores recibidos en un input, antes de pasarlos como parámetros a la siguiente etapa de ejecución. Aplica también a la impresión de valores almacenados en la BD, ya que en caso de almacenarse una instrucción maliciosa en la bd, la impresión del valor, como un código script, por ejemplo, pasaría al motor de ejecución sin ser ésta la intención del funcionamiento de la plataforma desarrollada.

SQL Injection.
Consiste en el paso malintencionado de comandos SQL al motor intérprete encargado de ejecutar dichos comandos sobre la base de datos, de esta manera, se pretende explotar la lógica de

programación creada para la consulta o almacenamiento de un valor, o almacenar instrucciones en la BD, para que sean ejecutadas al imprimir el valor en otra consulta.
Se pretende entonces, prevenir el agregar al dato en el input, un valor que siempre resulte verdadero, permitiendo a la consulta ejecutar la consulta independientemente de que el valor sea válido o existente, una variante consiste en añadir una instrucción SQL al valor, pretendiendo que se ejecute como si fuese parte de la consulta original, a modo de procesamiento por lotes de las instrucciones.

Consulta en código cgi	Valor esperado	Valor malintencionado	Resultado esperado	Resultado malintencionado
SELECT * FROM Users WHERE UserId =	105	105 OR 1=1	Dataset: John Smit.	Dataset: All Table Users
SELECT * FROM Users WHERE Name ="John Doe" AND Pass ="myPass"	John, Doe	" or ""=", "or ""="	Dataset: John Doe	Dataset: All Table Users
SELECT * FROM Users WHERE UserId =	105	105; DROP TABLE Suppliers	Dataset: John Smit.	Dataset: John Smit, Eliminacion de tabla Suppliers

Una buena práctica consiste en la parametrización del valor en la consulta interna codificada, evitando ejecuciones de SQL no intencionales. Otra buena práctica consiste en la limpieza de las cadenas pasadas como parámetros, eliminando caracteres de las cadenas de texto en el valor, que sean interpretados por el lenguaje sql, como operadores. Un ejemplo son los símbolos de operadores (%=;|"><$&+-/.?) AND, OR, LIKE.

Ataques XSS.
Como en el caso anterior, el ataque consiste en explotar la falta de validación de cadenas en los controles input de los formularios, para intentar pasar instrucciones de script activo, sean jquery, javascript, vbscript, e incluso instrucciones cgi (por ejemplo <?php echo ""?>), en la expectativa de que el siguiente motor interprete las instrucciones como parte del código original.
Un ejemplo de esto sería pasar una instrucción meta redirect de un control input, al almacenamiento en la BD de la plataforma, esta instrucción puede ser impresa en un listado, cuando el jquery intente darle formato a la lista, ejecutará también la instrucción impresa. Este comportamiento pudiera ser aprovechado para suplantar la plataforma legitima por una versión falsificada de la misma, intentando obtener valores que un usuario inadvertidamente introduzca, pensando que se encuentra en la plataforma original. Una prueba simple es introducir instrucciones echo o print en los controles de input para observar si poseen esta vulnerabilidad.
Para ambos casos, SQL injection y XSS Attack, siempre es recomendable que las validaciones a las cadenas de los controles input sean realizadas tanto del lado cliente como en el servidor que recibe el parámetro.

5. Content Security Police (CSP).

Las cabeceras de respuesta Content-Security-Policy, se presentan en la repuesta del servidor de la plataforma web, un cliente compatible reforzará la declaración de la política de lista blanca del navegador. Por ejemplo, se puede incluir en la política una ejecución de código JavaScipt mas estricta, tratando de prevenir ataques XSS, en realidad esta implementación conlleva una serie de características que son desactivadas por default:

- Código JavaScript inline.
  - Bloques <script>
  - Manejadores de evento DOM como atributos html (como por ejemplo .onclick)
  - Ligas url javascript: (en la barra url)
- Indicaciones de CSS inline.
  - Bloques <style>
  - Atributos style en elementos html
- Evaluación de código dinámico.
  - eval()
  - argumentos de texto para las funciones setTimeout y setInterval
  - el constructor new Function()
- Instrucciones CSS dinámicas.
  - El método CSSStylesheet.insertRule()

Desarrollar una plataforma nueva mientras se utiliza una CSP es un desafío, pero existen plataformas de desarrollo compatibles con CSP, es un hecho que muchas aplicaciones existentes deben cumplir rediseño o una política CSP no tan severa. La recomendación de buena práctica es utilizar el código css y script mediante las llamadas de carga de archivos externos (<script src>), interpretar vía json en vez de evaluarlos y utilizar EventTarget.addEventListener() para asignar manejadores de eventos.

6. Ocultamiento de mensajes de error y debug.

El ocultamiento de mensajes de error y de depuración, en una buena práctica por si misma, ya que ayuda a minimizar la información acerca de la plataforma en la que se ejecuta o se encuentra desarrollada la plataforma. De este modo dificulta al operador malintencionado, hacerse de información acerca de la versión, plataforma, web server o cgi sobre los cuales se ejecuta la aplicación.
Debe existir dentro del procedimiento de liberación de una plataforma web, una etapa que sea dedicada a la validación del ocultamiento de los mensajes de error tipo y mensajes de depuración. Si debe existir mensajes de error, deben ser los mensajes desarrollados para este fin específico, con mensajes propios del desarrollo y no del sistema.

7. Subida de archivos.

Se recomienda no utilizar métodos que permitan subir archivos al servidor mediante la plataforma web. Si no existe una alternativa, existen algunas consideraciones que se vuelven obligatorias en su implementación:

- Remoción de privilegios de ejecución del archivo.
- Renombramiento de las extensiones del archivo vía validación.
- Validación de existencia de comandos de ejecución en archivos de texto plano.

En la medida que algunos controles no puedan ser del todo implementado, se requiere considerar las implicaciones que conlleva la escritura de un archivo tipo scritp o shell en las carpetas del servidor, y de cómo esto representa una posibilidad latente de secuestro del servidor o permita otorgar puertas traseras al servidor mediante ejecución de comandos al nivel del sistema operativo.

8. Aislamiento de servidores de producción. DMZ

Es recomendable que la infraestructura de procesamiento se ejecute en un entorno que pueda ser monitoreado, que sus accesos puedan ser regulados y que se puedan determinar y clasificar intentos de conexiones no relacionadas al servicio que proporciona. Este entorno muchas veces se denomina DMZ (*demilitarized zone).* El concepto sugiere que los servidores puedan tener acceso a internet para la publicación de sus servicios, al mismo tiempo que el acceso de los mismos servidores hacia la red interna se encuentra restringido, de modo tal que, en caso de comprometer un servidor alojado en la dmz, no se compromete el acceso hacia los recursos de la red interna.

9. Uso de cifrado ssl.

El sitio debe seguir las recomendaciones para ser hospedado mediante el servicio de https, cada vez más máquinas de búsqueda e indexación favorecen en los resultados a los sitios que si cuentan con https sobre los que no lo hacen. Adicionalmente el servicio se ofrece cifrado de punto a punto entre el navegador visitante y la plataforma, dificultando la captura de datos al escuchar el medio de transporte.

10. Respaldos.

Se debe contar con un esquema que permita la recuperación en caso de incidentes sobre los componentes que soportan la plataforma web, en esencia, se debe contar con un esquema de respaldos que garanticen la integridad del código fuente, parámetros operativos del servidor, BD y webservice. Así como el control de las versiones de los elementos de librería utilizados. Se debe considerar respaldar, para recuperación en caso de falla:

- El código fuente de la plataforma.
- El código de las librerías y plataformas.
- Los instaladores del software de desarrollo.
- Bases de datos.
- Parámetros operativos de:
  - Sistema operativo

- Web Server
- Syslog
- Sistema de administración de BD.

11. Security checkups.

Se debe considerar la tarea de realizar security checkups sobre las plataformas desarrolladas, estas pruebas pueden realizarse en coordinación con el equipo de respuesta a incidentes, los resultados pueden arrojar posibles puntos de mejora, en caso de no ser aceptable el riesgo evaluado tras el uso de estos checkups.
Algunas plataformas que pueden utilizarse para la realización de los security checkups:

- Netsparker
- OpenVAS
- Security Headers – IO
- Xenotix XSS Exploit Framework.

(resultado de https://securityheaders.com/?q=www.yucatan.gob.mx&followRedirects=on)