**UNIVERSIDAD TECNOLÓGICA DE EL SALVADOR**

**FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS**

**ESCUELA DE INFORMÁTICA**


# ESTANDARES DE PROGRAMACIÓN

**UNIDAD: II**

**FACILITADOR: ING. EDWIN ALBERTO CALLEJAS**

**NOMBRE DE LA TAREA: DISEÑO DE SISTEMAS**


| ESTUDIANTES: | CARNET: | PART.: |
|---|---|---|
| ARCE AGUIRRE, JASON ALEXANDER | 25-0129-2017 | 100% |
| CASTILLO ALFARO, GABRIEL ALEJANDRO | 25-3339-2005 | 100% |
| CORDERO HERNANDEZ, KATHERINE ELIZABETH | 25-1461-2017 | 100% |
| LOVATO HUEZO, KEVIN ARNOLDO | 25-1642-2016 | 100% |
| MEJIA ORELLANA, DONOVAN ERNESTO | 25-1318-2014 | 100% |
| PORTILLO DELEON MIGUEL EDUARDO | 25-1596-2013 | 100% |
| SALAZAR MARTINEZ, JONATHAN OSWALDO | 25-6019-2016 | 100% |


**SEPTIEMBRE 24, 2020**

**INDICE**

# INTRODUCCION

Las organizaciones requieren transformar su negocio al siguiente nivel y avanzar a software más avanzado y especializado para beneficiarse de las tecnologías de vanguardia e renovar los modelos de negocio vigentes, para ello las empresas deben adaptarse y cambiar rápidamente desde adentro la forma en que desarrollan sus productos y servicios, para ello, el software es la clave de su transformación.

Otro factor importante para el diseño de los procesos es la necesidad de mitigar los riesgos del desarrollo de software, una parte considerable del riesgo del software se basa en procesos, por ejemplo, ha habido varios incidentes que podrían haberse evitado con estándares y herramientas de codificación adecuados, aunque estos estándares y herramientas están ampliamente disponibles, no se aplican o no se aplican adecuadamente en muchas situaciones debido a que esto normalmente se debe a la forma en que se organiza el trabajo, la seguridad física y lógica de los sistemas y a las personas con las que se lleva a cabo dicho trabajo. Es un problema del proceso de software y que las empresas deben encontrar, son formas de garantizar que los modelos de proceso se definan correctamente y que además se apliquen de manera adecuada

# OBJETIVOS

## GENERAL

Proporcionar un diseño al nivel del cambiante entorno técnico y de mercado proporcionando información que ayude a contruir de forma óptima y eficaz el software objetivo y enfatizar la necesidad considerarda en la evolución del proceso de desarrollo de software como un medio importante para la empresa.

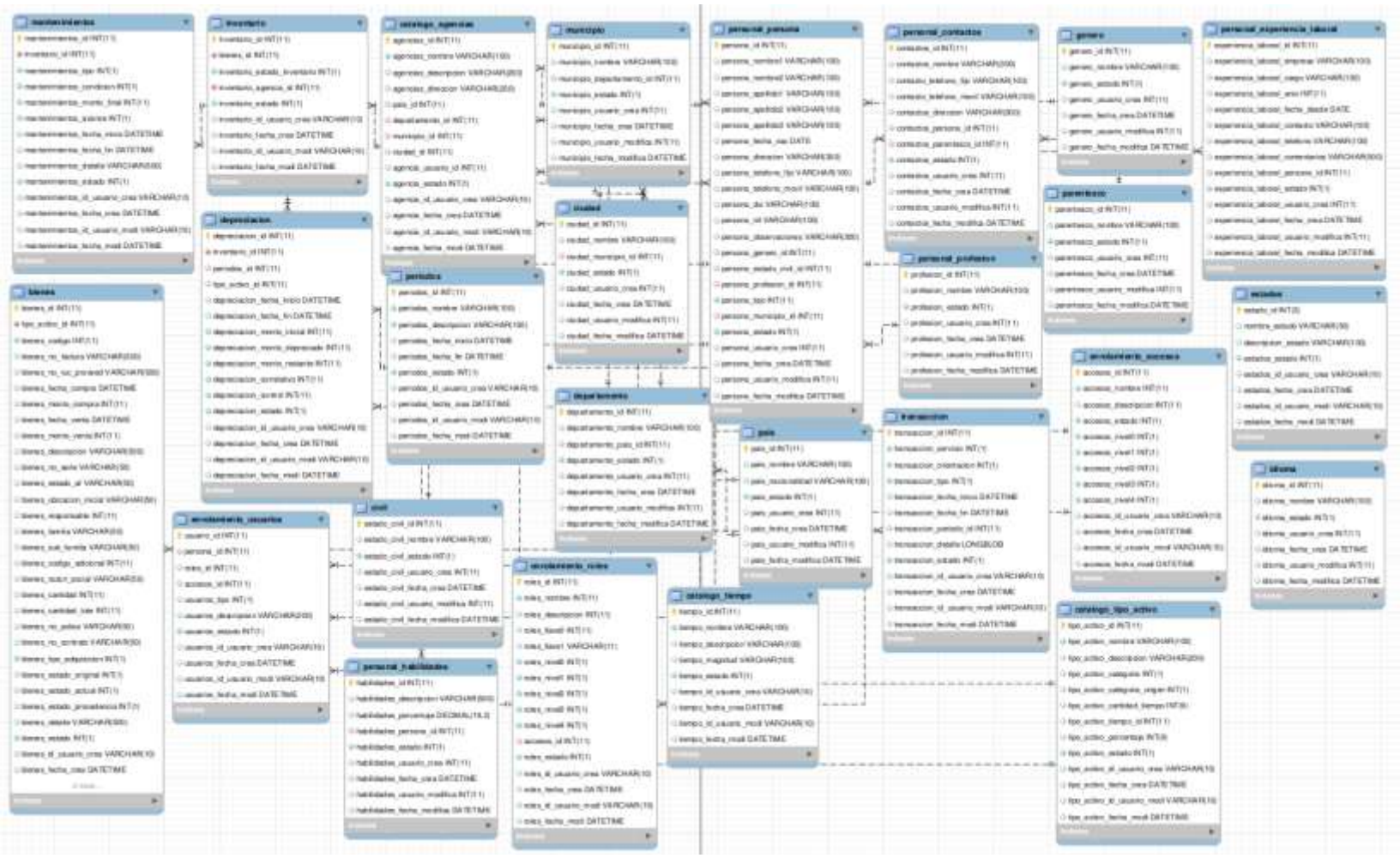
## ESPECIFICOS

Organizar el software de una organización y centrar en el diseño, desarrollo, gestión, gobernanza y aplicación del proceso de desarrollo de software en procesos que estén alineados con los objetivos comerciales de la empresa, como la expansión a nuevos dominios o el paso a la producción global.

Generar por medio de un entorno ágil el Sistema Transaccional de Activo Fijo para que pueda su construcción pueda ser gestionada por todos los miembros del equipo.

## DISEÑO DE DATOS

## DIAGRAMA RELACIONAL



**DICCIONARIO DE DATOS**

## TABLA: pais

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| pais_id | int | 11 | Llave primaria - Pais | 100000100001 |
| pais_nombre | varchar | 100 | Nombre | El Salvador |
| pais_nacionalidad | varchar | 100 | Nacionalidad/Gentilicio | Salvadoreno |
| pais_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| pais_usuario_crea | int | 11 | id del usuario que creo el registro | 200000100011 |
| pais_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| pais_usuario_modifica | int | 11 | id del usuario que modifico el registro | 200000100011 |
| pais_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: departamento

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| departamento_id | int | 11 | Llave primaria - Departamento | 100000100001 |
| departamento_nombre | varchar | 100 | Nombre | San Salvador |
| departamento_pais_id | int | 11 | Llave primaria - pais | 100000100001 |
| departamento_estado | int | 11 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 100000100001 |
| departamento_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| departamento_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| departamento_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| departamento_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: municipio

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| municipio_id | int | 11 | Llave primaria - Municipio | 100000100001 |
| municipio_nombre | varchar | 100 | nombre | San Salvador |
| municipio_departamento_id | int | 11 | Llave primaria - departamento | 100000100001 |
| municipio_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| municipio_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| municipio_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| municipio_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| municipio_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA:ciudad

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| ciudad_id | int | 11 | Llave primaria - Ciudad | 100000100001 |
| ciudad_nombre | varchar | 100 | nombre | San Salvador |
| ciudad_municipio_id | int | 11 | Llave primaria - municipio | 100000100001 |
| ciudad_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| ciudad_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| ciudad_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| ciudad_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| ciudad_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: genero

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| genero_id | int | 11 | Llave primaria - Genero | 100000100001 |
| genero_nombre | varchar | 100 | Nombre | Masculino |
| genero_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| genero_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| genero_fecha_crea datetime | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| genero_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| genero_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA:civil

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| estado_civil_id | int | 11 | Llave primaria - estado civil | 100000100001 |
| estado_civil_nombre | varchar | 100 | Nombre | Soltero |
| estado_civil_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| estado_civil_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| estado_civil_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| estado_civil_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| estado_civil_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA:idioma

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| idioma_id | int | 11 | Llave primaria - idioma | 100000100001 |
| idioma_nombre | varchar | 100 | Nombre | Espanol |
| idioma_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| idioma_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| idioma_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| idioma_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| idioma_fecha_modifica datetime | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA:parentesco

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| parentesco_id | int | 11 | Llave primaria - parentesco | 100000100001 |
| parentesco_nombre | varchar | 100 | Nombre | Madre |
| parentesco_estado | int | 1 | Estado de registro (-1= Eliminado, 0=inactivo, 1=activo) | 1 |
| parentesco_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| parentesco_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| parentesco_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| parentesco_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: personal_profesion

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| profesion_id | int | 11 | Llave primaria - profesion | 100000100001 |
| profesion_nombre | varchar | 100 | Nombre | Programador |
| profesion_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| profesion_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| profesion_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| profesion_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| profesion_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: personal_persona

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| persona_id i | int | 11 | Llave primaria - Persona | 100000100001 |
| persona_nombre1 | varchar | 100 | Llave primaria - Persona | Gabriel |
| persona_nombre2 | varchar | 100 | Nombre dos | Alejandro |
| persona_apellido1 | varchar | 100 | Apellido uno | Castillo |
| persona_apellido2 | varchar | 100 | Apellido dos | Alfaro |
| persona_apellido3 | varchar | 100 | Apellido tres (casada) | null |
| persona_fecha_nac | date | | Fecha de nacimiento | 2020/09/24 |
| persona_direccion | varchar | 300 | Dirección | Direccion completa del contacto |
| persona_telefono_fijo | varchar | 100 | Telefono fijo | 987654321 |
| persona_telefono_movil | varchar | 100 | Telefono movil | 987654321 |
| persona_dui | varchar | 100 | Dui | 12345677 |
| persona_nit | varchar | 100 | nit | 06110000001239 |
| persona_observaciones | varchar | 300 | Dui | 12345677 |
| persona_genero_id | int | 11 | Llave primaria - Genero | 100000100001 |
| persona_estado_civil_id | int | 11 | Llave primaria - estado civil | 100000100001 |
| persona_profesion_id | int | 11 | Llave primaria - profesion | 100000100001 |
| persona_tipo | int | 1 | (0=contratado, 1=temporal, 2=outsourcing, 3=facilitador, 4=externo, 5=proveedor, 6=cliente) | 1 |
| persona_municipio_id | int | 11 | Llave primaria - municipio | 100000100001 |
| persona_estado | int | 1 | Estado de registro (0=inactivo, 1=activo, 2= Eliminado) | 1 |
| persona_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| persona_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| persona_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| persona_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: personal_habilidades

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| habilidades_id | int | 11 | Llave primaria - Habilidades | 100000100001 |
| habilidades_descripcion | varchar | 500 | Descripción habilidad | Descripcion de la habilidad |
| habilidades_porcentaje | decimal | 19,2 | Porcentaje sobre la habilidad | 15 |
| habilidades_persona_id | int | 11 | Llave primaria - Persona | 100000100001 |
| habilidades_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| habilidades_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| habilidades_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| habilidades_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| habilidades_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: personal_experiencia_laboral

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| experiencia_laboral_id | int | 11 | Llave primaria - Experiencia Labora | 100000100001 |
| experiencia_laboral_empresa | varchar | 100 | Empresa | Nombre de la Empresa |
| experiencia_laboral_cargo | varchar | 100 | Cargo | Cargo en la Empresa |
| experiencia_laboral_anio | int | 11 | Anios de experiencia | 100000100001 |
| experiencia_laboral_fecha_desde | date | | Fecha desde | 2020/09/24 |
| experiencia_laboral_contacto | varchar | 100 | Contacto | Persona de Contacto |
| experiencia_laboral_telefono | varchar | 100 | Telefono | 9876543211 ext 1234 |
| experiencia_laboral_comentarios | varchar | 500 | Comentarios | Comentarios adicionales |
| experiencia_laboral_persona_id | int | 11 | Llave primaria - Persona | 100000100001 |
| experiencia_laboral_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| experiencia_laboral_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| experiencia_laboral_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| experiencia_laboral_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| experiencia_laboral_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: enrolamiento_accesos

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| accesos_id | int | 11 | | 100000100001 |
| accesos_nombre | int | 11 | | 100000100001 |
| accesos_descripcion | int | 11 | | 100000100001 |
| accesos_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| accesos_nivel0 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| accesos_nivel1 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| accesos_nivel2 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| accesos_nivel3 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| accesos_nivel4 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| accesos_id_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| accesos_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| accesos_id_usuario_modi | int | 11 | id del usuario que modifico el registro | 100000100001 |
| accesos_fecha_modi datetime | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## TABLA: personal_contactos

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCION | EJEMPLO |
|---|---|---|---|---|
| contactos_id | int | 11 | Llave primaria - contactos | 100000100001 |
| contactos_nombre | varchar | 300 | Nombre completo del contacto | Nombre Completo del Contaco |
| contacto_telefono_fijo | varchar | 100 | Telefono fijo | 9877654321 |
| contacto_telefono_movil | varchar | 100 | Telefono movil | 987654321 |
| contactos_direccion | varchar | 300 | Dirección del contacto | Direccion completa del contacto |
| contactos_persona_id | int | 11 | Llave primaria - Persona | 100000100001 |
| contactos_parentesco_id | int | 11 | Llave primaria - Parentesco | 100000100001 |
| contactos_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| contactos_usuario_crea | int | 11 | id del usuario que creo el registro | 100000100001 |
| contactos_fecha_crea | datetime | | fecha de creacion del registro | 2020/09/24 10:00:01 |
| contactos_usuario_modifica | int | 11 | id del usuario que modifico el registro | 100000100001 |
| contactos_fecha_modifica | datetime | | fecha de modificacion del registro | 2020/09/24 10:00:01 |

## Tabla: bienes

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| bienes_id | int | 11 | | 100000100001 |
| tipo_activo_id | int | 11 | | 100000100001 |
| bienes_codigo | int | 11 | | 100000100001 |
| bienes_no_factura | varchar | 500 | Numero de factura | 100000100001 |
| bienes_no_ruc_proveed | varchar | 500 | Numero ruc proveedor | 100000100001 |
| bienes_fecha_compra | datetime | | | 24/09/2020 10:00 |
| bienes_monto_compra | int | 11 | | 100000100001 |
| bienes_fecha_venta | datetime | | | 24/09/2020 10:00 |
| bienes_monto_venta | int | 11 | | 100000100001 |
| bienes_descripcion | varchar | 500 | | Inmueble |
| bienes_no_serie | varchar | 50 | Numero de serie | 111 |
| bienes_estado_af | varchar | 50 | Estado af | En buen estado |
| bienes_ubicacion_inicial | varchar | 50 | Ubicacion fisica inicial del bien | San Salvador |
| bienes_responsable | int | 11 | Usuario responsable inicialmente del bien | Juan Paredes |
| bienes_familia | varchar | 50 | familia a que pertenece | edificios |
| bienes_sub_familia | varchar | 50 | familia a que pertenece | edificios |
| bienes_codigo_adicional | int | 11 | Codigo adicional de identificacion | 100000100001 |
| bienes_razon_social | varchar | 50 | Razon social | Salud |
| bienes_cantidad | int | 11 | | 100000100001 |
| bienes_cantidad_lote | int | 11 | | 100000100001 |
| bienes_no_poliza | varchar | 50 | Numero de poliza | 1111 |
| bienes_no_contrato | varchar | 50 | Numero del contrato | 2222 |
| bienes_tipo_adquisicion | int | 1 | 1=compra directa, 2=arrendamiento temporal, 3=arrendamiento contratado, 4=compra por arrendamiento, 5=servicio contratado | 3 |

| | | | | |
|---|---|---|---|---|
| bienes_estado_original | int | 1 | 1=unidad completa, 2=parte de una unidad, 3=resultado de desmantelamiento, 4=producto de otro bien, 5=generado uniendo bienes | 4 |
| bienes_estado_actual | int | 1 | 1=unidad completa, 2=parte de una unidad, 3=resultado de desmantelamiento, 4=producto de otro bien, 5=generado uniendo bienes | 2 |
| bienes_estado_procedencia | int | 1 | 1=unidad completa, 2=parte de una unidad, 3=resultado de desmantelamiento, 4=producto de otro bien, 5=generado uniendo bienes | 5 |
| bienes_detalle | varchar | 500 | | Segunda planta |
| bienes_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo | 0 |
| bienes_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 4214 |
| bienes_fecha_crea | datetime | | fecha de creacion del registro | 24/09/2020 10:00 |
| bienes_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 3231 |
| bienes_fecha_modi | datetime | | fecha de modificacion del registro | 24/09/2020 10:00 |

## TABLA: depreciacion

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| depreciacion_id | int | 11 | | 100000100001 |
| inventario_id | int | 11 | | 100000100001 |
| periodos_id | int | 11 | | 100000100001 |
| tipo_activo_id | int | 11 | | 100000100001 |
| depreciacion_fecha_inicio | datetime | | Fecha de inicio del calculo de depreciacion | 24/09/2020 10:00 |
| depreciacion_fecha_fin | datetime | | Fecha de inicio del calculo de depreciacion | 25/09/2020 10:00 |
| depreciacion_monto_inicial | int | 11 | | 100000100001 |
| depreciacion_monto_depreciado | int | 11 | | 100000100001 |
| depreciacion_monto_restante | int | 11 | | 100000100001 |
| depreciacion_correlativo | int | 11 | | 100000100001 |
| depreciacion_control | int | 11 | | 100000100001 |
| depreciacion_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo | 2 |
| depreciacion_id_usuario crea | varchar | 10 | id del usuario que creo el registro | 4526 |
| depreciacion_fecha_crea | datetime | | fecha de creacion del registro | 25/09/2020 10:00 |
| depreciacion_id_usuario modi | varchar | 10 | id del usuario que modifico el registro | 3324 |
| depreciacion_fecha_modi | datetime | | fecha de modificacion del registro | 25/09/2020 10:00 |

## TABLA: mantenimientos

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| mantenimientos_id | int | 11 | | 100000100001 |
| inventario_id | int | 11 | | 100000100001 |
| mantenimientos_tipo | int | 1 | 1=programado, 2=solicitado 3=preventivo, 4=correctivo | 4 |
| mantenimientos_condicion | int | 1 | 1=incluido, 2=contratado 3=licitado, 4=emergencia | 1 |
| mantenimientos_monto_final | int | 11 | Monto por el cual se esta efectuando el mantenimiento | 100000100001 |
| mantenimientos_avance | int | 1 | 1=finalizado, 2=incompleto 3=reclamo, 4=reprogramacion | 2 |
| mantenimientos_fecha_inicio | datetime | | fecha de inicio del mantenimiento | 25/09/2020 10:00 |
| mantenimientos_fecha_fin | datetime | | fecha de finalizacion del mantenimiento | 26/09/2020 10:00 |
| mantenimientos_detalle | varchar | 100 | Detalles adicionales del mantenimiento | Se necesitan nuevas herramientas |
| mantenimientos_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 2 |
| mantenimientos_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 5577 |
| mantenimientos_fecha_crea | datetime | | fecha de creacion del registro | 26/09/2020 10:00 |
| mantenimientos_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 8832 |
| mantenimientos_fecha_modi | datetime | | fecha de modificacion del registro | 26/09/2020 10:00 |

## TABLA: inventario

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| inventario_id | int | 11 | | 100000100001 |
| bienes_id | int | 11 | | 100000100001 |
| inventario_estado_inventario | int | 1 | Estado de Activo Fijo 0=creado, 1=activo, 2=obsoleto, 3=baja del inventario, 4=venta del activo | 3 |
| inventario_agencia_id | int | 11 | Agencia en donde se encuentra fisicamente el activo | Agencia Morales |
| inventario_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| inventario_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 3326 |
| inventario_fecha_crea | datetime | | fecha de creacion del registro | 26/09/2020 10:00 |
| inventario_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 3523 |
| inventario_fecha_modi | datetime | | fecha de modificacion del registro | 26/09/2020 10:00 |

## TABLA: transaccion

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| transaccion_id | int | 11 | | 100000100001 |
| transaccion_servicio | int | 1 | | 1 |
| transaccion_orientacion | int | 1 | 1=Request, 2=Response | 2 |
| transaccion_tipo | int | 1 | 1=Creacion, 2=Recuperacion, 3=Modificacion, 4=Borrar, 5=logueo al sistema, 6=salida del sistema | 6 |
| transaccion_fecha_inicio | datetime | | fecha de inicio de la transaccion | 26/09/2020 10:00 |
| transaccion_fecha_fin | datetime | | fecha de finalizacion de la transaccion | 27/09/2020 10:00 |
| transaccion_periodo_id | int | 11 | | 100000100001 |
| transaccion_detalle | LONGBLOB | | | 3413331 |
| transaccion_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 0 |
| transaccion_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 7362 |
| transaccion_fecha_crea | datetime | | fecha de creacion del registro | 27/09/2020 10:00 |
| transaccion_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 2892 |
| transaccion_fecha_modi | datetime | | fecha de modificacion del registro | 27/09/2020 10:00 |

## TABLA: periodos

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| periodos_id | int | 11 | | 100000100001 |
| periodos_nombre | varchar | 100 | | periodo segundo |
| periodos_descripcion | varchar | 100 | | abril, mayo |
| periodos_fecha_inicio | datetime | | fecha de inicio del periodo | 27/09/2020 10:00 |
| periodos_fecha_fin | datetime | | fecha de finalizacion del periodo | 28/09/2020 10:00 |
| periodos_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 2 |
| periodos_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 2382 |
| periodos_fecha_crea | datetime | | fecha de creacion del registro | 27/09/2020 10:00 |
| periodos_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 3672 |
| periodos_fecha_modi | datetime | | fecha de modificacion del registro | 27/09/2020 10:00 |

## TABLA: catalogo_tiempo

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| tiempo_id | int | 11 | | 100000100001 |
| tiempo_nombre | varchar | 100 | | domingo |
| tiempo_descripcion | varchar | 100 | | mañana |
| tiempo_magnitud | varchar | 100 | 0=anio, 1=semestre, 2=trimestre, 3=bimestre, 4=mensual, 5=quincenal, 6=semanal, 7=diario | 5 |
| tiempo_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 0 |
| tiempo_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 3222 |
| tiempo_fecha_crea | datetime | | fecha de creacion del registro | 27/09/2020 10:00 |
| tiempo_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 3413 |
| tiempo_fecha_modi | datetime | | fecha de modificacion del registro | 27/09/2020 10:00 |

## TABLA: catalogo_tipo_activo

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| tipo_activo_id | int | 11 | | 100000100001 |
| tipo_activo_nombre | varchar | 100 | | edificio |
| tipo_activo_descripcion | varchar | 200 | | hospital |
| tipo_activo_categoria | int | 1 | 0=activo principal ,1=subcategoria, 2=extraido de un activo ppal 3=extraido de una subcategoria | 1 |
| tipo_activo_categoria_origen | int | 1 | 0=activo principal ,1=subcategoria, 2=extraido de un activo ppal 3=extraido de una subcategoria | 2 |
| tipo_activo_cantidad_tiempo | int | 9 | | 7 |
| tipo_activo_tiempo_id | int | 11 | Factor por el cual se efectuara la medicion de la vida util o depreciacion | 100000100001 |
| tipo_activo_porcentaje | int | 9 | Porcentaje sobre valor por escala de tiempo | 9 |
| tipo_activo_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 2 |
| tipo_activo_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 1323 |
| tipo_activo_fecha_crea | datetime | | fecha de creacion del registro | 27/09/2020 10:00 |
| tipo_activo_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 2812 |
| tipo_activo_fecha_modi | datetime | | fecha de modificacion del registro | 27/09/2020 10:00 |

## TABLA: catalogo_agencias

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| agencias_id | int | 11 | | 100000100001 |
| agencias_nombre | varchar | 100 | | agencia morales |
| agencias_descripcion | varchar | 200 | | agencia activa |
| agencias_direccion | varchar | 200 | | San Salvador |
| pais_id | int | 11 | | 100000100001 |
| departamento_id | int | 11 | | 100000100001 |
| municipio_id | int | 11 | | 100000100001 |
| ciudad_id | int | 11 | | 100000100001 |
| agencia_usuario_id | int | 11 | Usuario encargado de la agencia | 100000100001 |
| agencia_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 0 |
| agencia_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 3423 |
| agencia_fecha_crea | datetime | | fecha de creacion del registro | 27/09/2020 10:00 |
| agencia_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 3523 |
| agencia_fecha_modi | datetime | | fecha de modificacion del registro | 27/09/2020 10:00 |

## TABLA: estados

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| estado_id | int | 3 | id del estado - tabla estados | 343 |
| nombre_estado | varchar | 50 | nombre a mostrar en el sistema | financiero |
| descripcion_estado | varchar | 100 | Define el tipo de aplicacion del estado a la tabla | activo |
| estados_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 2 |
| estados_id_usuario_crea | varchar | 10 | id del usuario que creo el estado | 3237 |
| estados_fecha_crea | datetime | | fecha de creacion del estado | 27/09/2020 10:00 |
| estados_id_usuario_modi | varchar | 10 | id del usuario que modifico el estado | 3248 |
| estados_fecha_modi | datetime | | fecha de modificacion del estado | 27/09/2020 10:00 |

## TABLA: enrolamiento_usuarios

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| usuario_id | int | 11 | | 100000100001 |
| persona_id | int | 11 | | 100000100001 |
| roles_id | int | 11 | | 100000100001 |
| accesos_id | int | 11 | | 100000100001 |
| usuarios_tipo | int | 1 | | 4 |
| usuarios_descripcion | varchar | 200 | | director |
| usuarios_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| usuarios_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 6423 |
| usuarios_fecha_crea | datetime | | fecha de creacion del registro | 27/09/2020 10:00 |
| usuarios_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 3829 |
| usuarios_fecha_modi | datetime | | fecha de modificacion del registro | 27/09/2020 10:00 |

## TABLA: enrolamiento_roles

| CAMPO | TIPO DE DATO | TAMAÑO | DESCRIPCIÓN | EJEMPLO |
|---|---|---|---|---|
| roles_id | int | 11 | | 100000100001 |
| roles_nombre | int | 11 | | 100000100001 |
| roles_descripcion | int | 11 | | 100000100001 |
| roles_llave0 | int | 11 | | 100000100001 |
| roles_llave1 | int | 11 | | 100000100001 |
| roles_nivel0 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 2 |
| roles_nivel1 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 0 |
| roles_nivel2 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| roles_nivel3 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 2 |
| roles_nivel4 | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 0 |
| accesos_id | int | 11 | | 100000100001 |
| roles_estado | int | 1 | Estado de registro (2= Eliminado, 0=inactivo, 1=activo) | 1 |
| roles_id_usuario_crea | varchar | 10 | id del usuario que creo el registro | 4534 |
| roles_fecha_crea | datetime | | fecha de creacion del registro | 27/09/2020 10:00 |
| roles_id_usuario_modi | varchar | 10 | id del usuario que modifico el registro | 4645 |
| roles_fecha_modi | datetime | | fecha de modificacion del registro | 27/09/2020 10:00 |

## DISEÑO DE INTERFACES

## PATRONES DE DISEÑO

Los patrones de diseño ofrecen soluciones a problemas comunes de diseño de aplicaciones y en la programación orientada a objetos, los patrones de diseño normalmente están dirigidos a resolver los problemas asociados con la creación e interacción de objetos, en lugar de los problemas a gran escala que enfrenta la arquitectura general del software, proporcionando soluciones generalizadas en forma de codigo repetitivo que se puede aplicar a problemas de la vida real en general se visualizan mediante un diagrama de clases, que muestra los comportamientos y las relaciones entre las clases.

Los patrones de diseño son infinitamente útiles y soluciones comprobadas a los problemas que inevitablemente enfrentará. No solo imparten años de conocimiento y experiencia colectivos, los patrones de diseño ofrecen un buen vocabulario entre los desarrolladores y arrojan luz sobre muchos problemas.

Sin embargo, los patrones de diseño no son una varita mágica; no ofrecen una implementación lista para usar como un marco o un conjunto de herramientas. El uso innecesario de patrones de diseño, solo porque suenan geniales o porque desea impresionar, puede resultar en un sistema sofisticado y con una ingeniería excesiva que no resuelve ningún problema sino que introduce errores, diseño ineficaz, bajo rendimiento y problemas de mantenimiento, la mayoría de los patrones pueden resolver problemas de diseño, proporcionar soluciones fiables a problemas conocidos y permitir a los desarrolladores comunicarse en un idioma común para todos y solo deben usarse cuando es probable que ocurran problemas.

### Patrón FACTORY

El patrón de diseño Factory es uno de los patrones de diseño básicos más utilizados en los lenguajes de programación modernos y utilizado no solo por los desarrolladores web y de aplicaciones, sino también los desarrolladores de tiempos de ejecución y frameworks como Java y Spring.

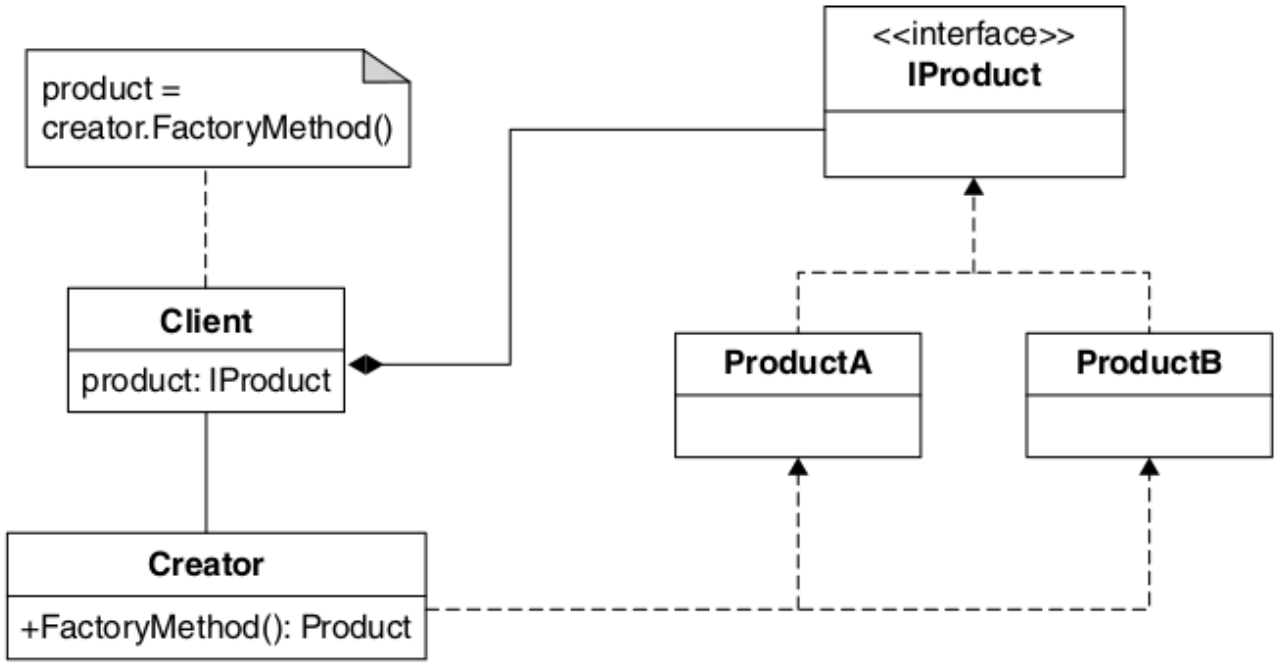
Este patrón tiene dos variaciones: el método factory y abstract factory. La intención es la misma: proporcionar una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas.

Siendo un caso concreto como en el presente sistema transaccional, necesitamos insertar datos en nuestra base de datos, primero tenemos que crear una SqlConnection y solo entonces podremos continuar. Si los ponemos en simples if-else, necesitamos repetir mucho código y no quedara bien. En

este momento podemos utilizar el patrón Factory para resolver este tipo de problemas, la estructura básica se define con una clase abstracta y nuestras subclases se derivarán de esta clase, siendo que las subclases asumirán la responsabilidad del proceso de creación de instancias.

El patrón Factory da la oportunidad de desacoplar la creación de objetos del sistema subyacente al encapsular el código responsable de la creación de los objetos. Este enfoque simplifica nuestra vida cuando se trata de refactorizar, ya que ahora tenemos un solo punto donde ocurren los cambios de refactorización. A menudo, el Factory se implementa como un singleton o como una clase estática porque normalmente solo se requiere una instancia del Factory, esto centraliza la creación de objetos de Factory, lo que permite una mayor organización y mantenimiento del código fuente y la reducción de errores cuando se realizan cambios y actualizaciones.

A continuación, un ejemplo de cómo la creación del objeto está encapsulada en las subclases:




**Patrón FAÇADE**

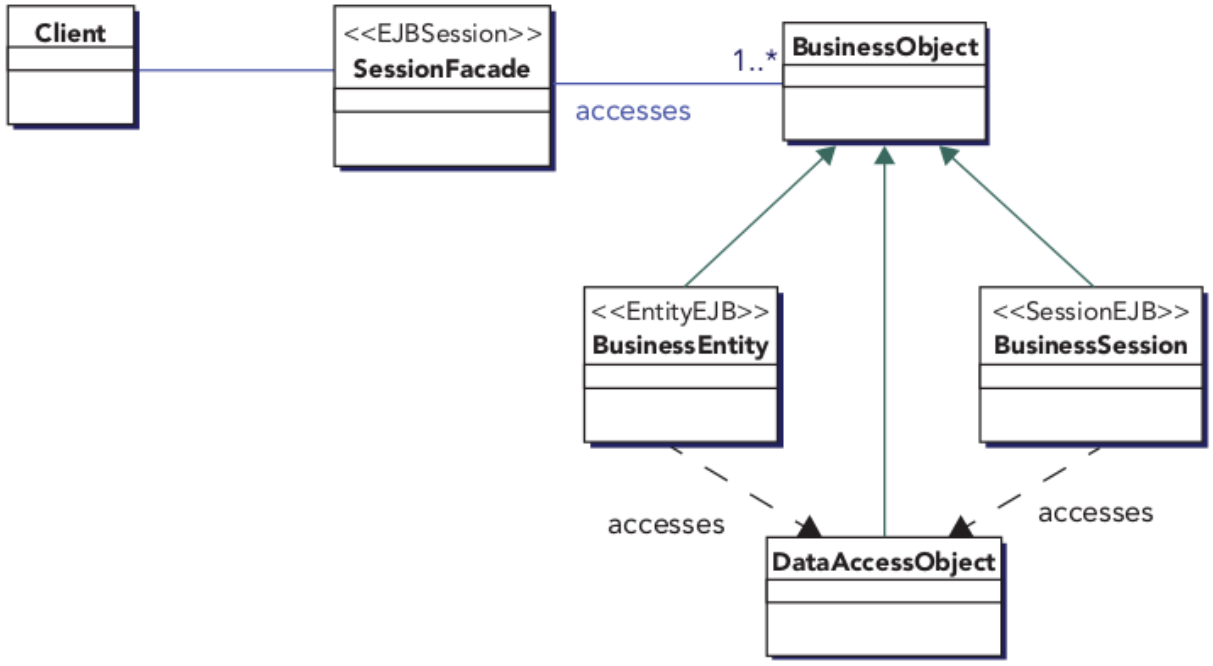
El patrón de la façade es uno de los patrones de diseño estructural descritos en el libro GoF 1. La intención detrás de esto y lo que se hara con la implementacion del presente sistema transaccional de activo fijo y combinado con el patron de Factory, es encapsular la lógica empresarial complicada en una interfaz de nivel superior que hace que el acceso a un subsistema sea más fácil de usar. Esto se hace a menudo agrupando las llamadas a métodos relacionados e invocándolas secuencialmente desde un método.

Desde una vista de nivel superior, cada API puede considerarse una implementación del patrón de façade, ya que proporcionan una interfaz simple que oculta su complejidad. Cualquier llamada al método de una API da como resultado la invocación de muchos otros métodos desde un subsistema oculto detrás de él. Un ejemplo de fachada sería la interfaz javax.servlet.http.HttpSession. Esto esconde la complicada lógica asociada con el mantenimiento de la sesión al tiempo que expone su funcionalidad a través de un puñado de métodos fáciles de usar.

El patrón façade se implementa comúnmente para los siguientes propósitos y situaciones:

- Proporcione un acceso simple y unificado a un sistema de back-end heredado.
- Cree una API pública para clases, como un controlador.
- Ofrecer acceso al coarse-grained a los servicios disponibles y los servicios se combinan.
- Reduce las llamadas en la red ya que Façade realiza muchas llamadas al subsistema, mientras que el cliente remoto realiza una sola llamada al façade.
- Para encapsular el flujo y los detalles internos de una aplicación para mayor seguridad y simplicidad.

Como se puede ver en el siguiente diagrama, el patrón de fachada proporciona una interfaz simple a un sistema subyacente. Encapsula la complicada lógica granular.



**Patrón MVC**

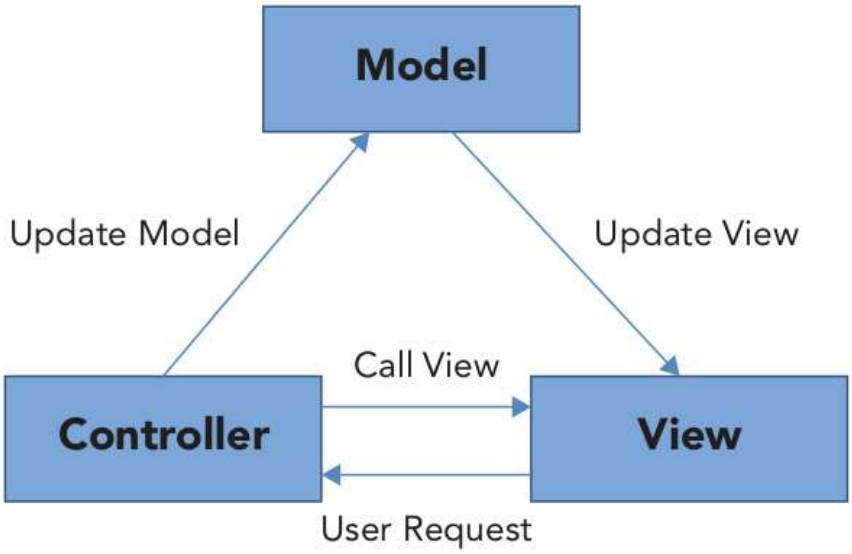
El patrón de Modelo Vista Controlador (MVC) es uno de los patrones de diseño arquitectónico más ubicuos en el desarrollo de aplicaciones modernas que se enumera en el libro de Gang of Four. Se basa en la filosofía de separación de preocupaciones y encapsula el procesamiento de los datos de la aplicación desde la presentación de los datos. No encapsula el procesamiento de datos a partir de la presentación de los datos, conduce a sistemas altamente acoplados que son difíciles de mantener y ampliar. La separación de preocupaciones que proporciona el patrón MVC hace que las modificaciones tanto en la lógica empresarial como en la interfaz de usuario sean mucho más fáciles e independientes.

En MVC, el modelo representa los datos de la aplicación y la lógica empresarial relacionada. El modelo puede estar representado por un objeto o un gráfico complejo de objetos relacionados. En una aplicación Java EE, los datos se encapsulan en objetos de dominio que a menudo se implementan en un módulo EJB. Los datos se transportan hacia y desde la capa de acceso a la base de datos en objetos de transferencia de datos (DTO) y se accede a ellos a través de objetos de acceso a datos (DAO).

La vista es la representación visual de los datos contenidos en el modelo, un subconjunto del modelo se representa en una sola vista; por lo tanto, la vista actúa como un filtro para los datos del modelo. El usuario interactúa con los datos del modelo a través de la representación visual de la vista e invoca la lógica empresarial que, a su vez, actúa sobre los datos del modelo.

El controlador vincula la vista con el modelo y dirige el flujo de la aplicación, elige qué vista mostrar al usuario en respuesta a la entrada del usuario y la lógica empresarial que se procesa. El controlador recibe un mensaje de la vista, que reenvía al modelo. El modelo, a su vez, prepara una respuesta y la envía de vuelta al controlador donde se elige la vista y se envía al usuario.

Lógicamente abarca el cliente y el nivel medio de una arquitectura de varios niveles. En un entorno Java EE, el modelo está ubicado en la capa empresarial, normalmente en forma de un módulo Enterprise JavaBeans (EJB). El controlador y la vista se encuentran en el nivel web. Es probable que la vista se construya a partir de JavaServer Faces (JSF) o JavaServer Pages (JSP) con la ayuda de Expression Language (EL). El controlador es normalmente un servlet que recibe solicitudes de protocolo de transferencia de hipertexto (HTTP) del usuario.




El patrón MVC viene en formas diferentes y los dos más reconocidos se conocen como Tipo I y Tipo II, siendo este ultimo el propuesto a utilizarse en nuestro sistema:

MVC Tipo I: este tipo es un enfoque centrado en la página, en el que la vista y el controlador existen como una entidad denominada controlador de vista. Con este enfoque, la lógica del controlador se implementa dentro de la vista, como en un JSF. Todas las tareas que realiza el controlador, incluida la recuperación de los atributos y parámetros de la solicitud HTTP, la invocación de la lógica empresarial y la gestión de la sesión HTTP, están integradas en la vista mediante scriptlets y bibliotecas de etiquetas. El tipo I combina en gran medida la generación de vistas con el flujo de la aplicación, lo que dificulta el mantenimiento.

mple

MVC Tipo II: los problemas de mantenimiento con el Tipo I se superan en el Tipo II moviendo la lógica del controlador fuera de la vista y dentro de un servlet, dejando que JSF se ocupe de la representación de los datos para la vista.

**Patrón DATA ACCESS**

La forma en que utiliza las fuentes de datos puede variar sustancialmente y su implementación puede diferir mucho ya que actualmente hay diferentes dialectos de SQL, como Postgre SQL y Oracle. El objetivo simple del patrón de objeto de acceso a datos (DAO) es encapsular el acceso a la fuente de datos proporcionando una interfaz a través de la cual las diversas capas pueden comunicarse con la fuente de datos, por tanto, es inimaginable pensar en una aplicación empresarial que no interactúe de alguna manera con las fuente de datos, esta puede ser una base de datos relacional, orientada a objetos o NoSQL, un repositorio de Protocolo ligero de acceso a directorios (LDAP), un sistema de archivos, un servicio web o un sistema externo, sea cual fuere la fuente de los datos, la aplicación empresarial debe interactuar con ellos y realizar operaciones básicas de creación, recuperación, modificación y eliminación (CRUD). Casi todos los servidores utilizan dichas fuentes de datos para mantener sesiones o procesos de larga ejecución sin problemas.

Se pensó que si cambiaba la fuente de datos, el desacoplamiento reduciría o anularía cualquier impacto. Sin embargo, en realidad, la fuente de datos rara vez cambia, ni siquiera entre proveedores del mismo tipo de fuente, como entre Postgre y MS SQL. Es difícil imaginar que se tomaría la decisión de migrar una fuente de datos SQL a un sistema de archivos XML, repositorio LDAP o servicio web.

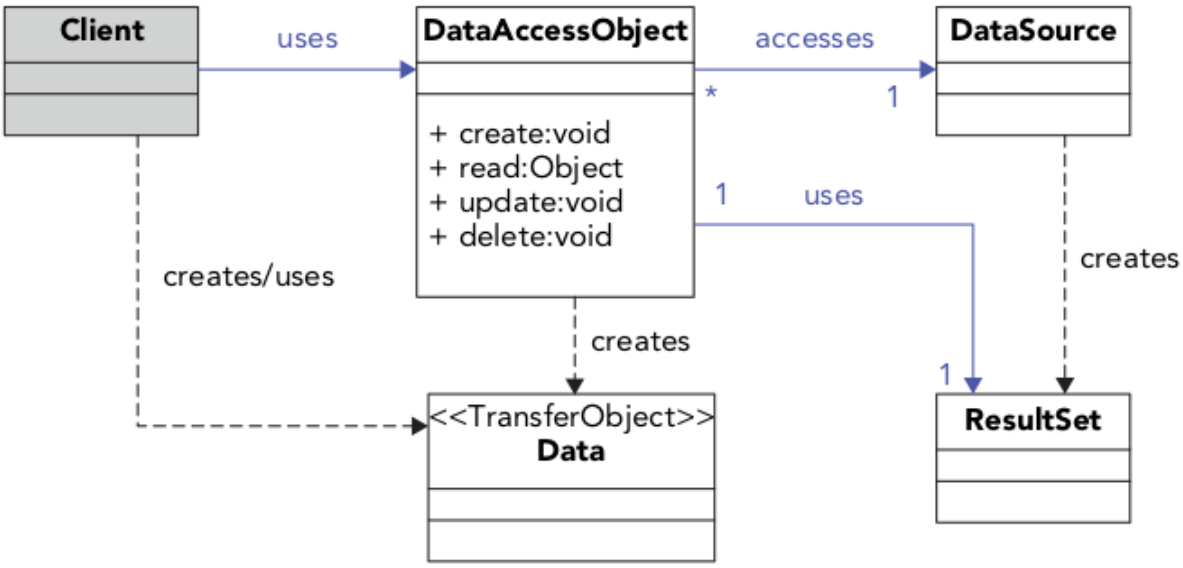
El patrón DAO sigue siendo un patrón valioso, y su solución original sigue siendo válida, aunque la motivación para su implementación ha cambiado en su énfasis. En lugar de protegerse contra el impacto de un cambio improbable en el tipo de fuente de datos, el valor está en su similitud y capacidad de prueba y su uso para estructurar el código y mantenerlo limpio de código de acceso a datos. Todavía tiene valor usarlo como una forma de encapsular sistemas de almacenamiento de datos heredados y simplificar el acceso a implementaciones complejas de fuentes de datos. Sin embargo, es más probable que se trate de casos excepcionales.

El patrón DAO encapsula las operaciones CRUD en una interfaz que es implementada por una clase concreta. Esta interfaz se puede burlar y, por lo tanto, probar fácilmente, evitando una conexión a la base de datos. Las pruebas se han mejorado porque se usa el UnitTesting con herramientas Mock siendo mas faciles de integrar que con una base de datos en vivo. El implementacion concreta de DAO utiliza un API de bajo nivel como JPA e Hibernate para realizar las operaciones CRUD.

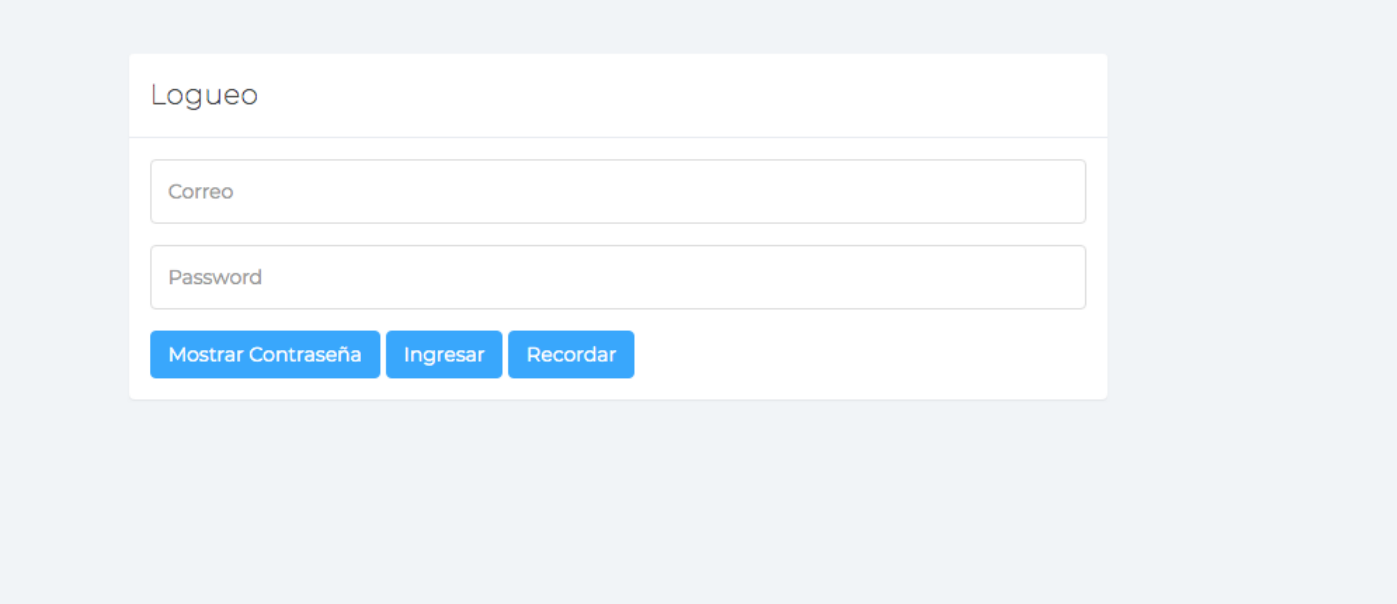
La implementación del patrón DAO involucra varios componentes:

- La interfaz DAO
- La implementación concreta de la interfaz DAO
- La fábrica de DAO
- El DTO

A continuacion, el diagrama de clases del patrón Data Access:

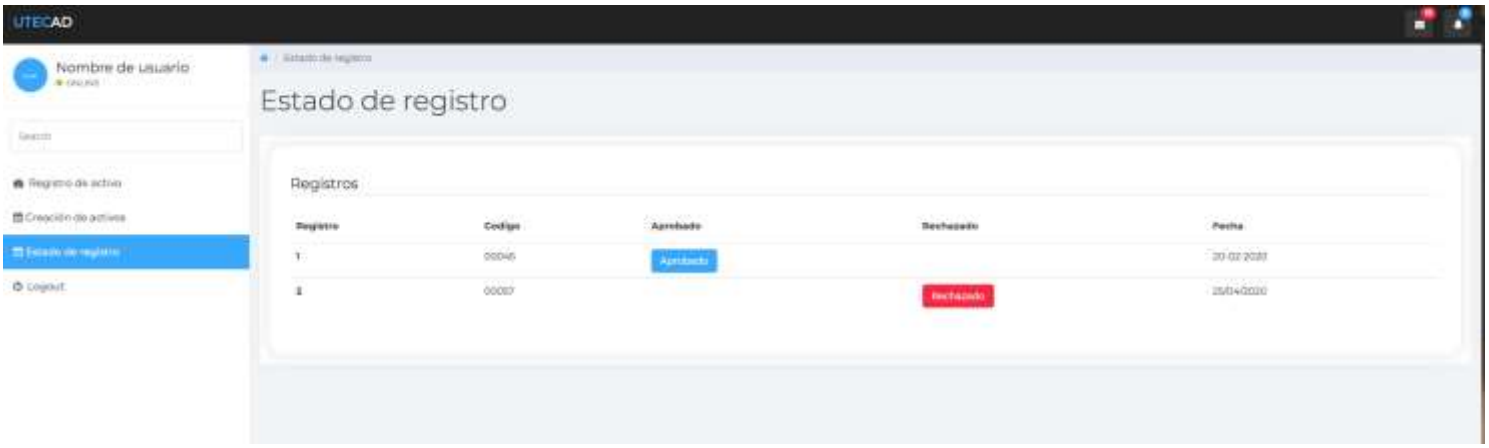


## DISEÑO DE INTERFACES DE ENTRADA, SALIDA Y REPORTES

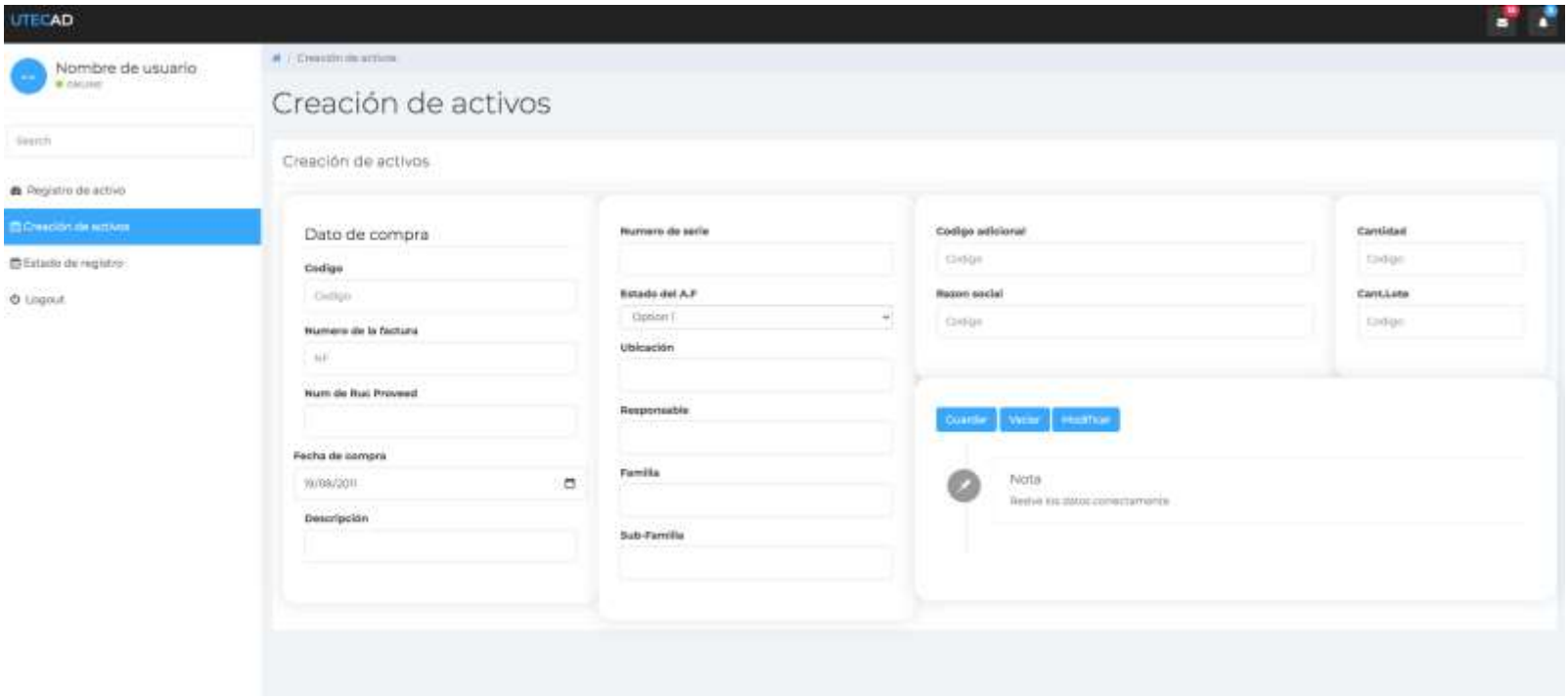



| | | | | | |
|---|---|---|---|---|---|
| Pantalla: | REGISTRO DE SESION | | | Interfaz | 1 |
| Descripcion: | Pantalla generica de registro y salida de sesion del sistema | | | | |
| Objetivo: | Proporcionar las credenciales requeridas para un inicio de sesion exitoso y permite salir ordenadamente del sistema apareciendo al desloguearse del mismo. | | | | |
| Usuarios: | Aplica para todos los usuarios del sistema. | | | | |
| Datos de Entrada | | | | | |
| Detalle | Tipo | | | | |
| | Digitar | Seleccionar | Recuperar | Generar | Obligatorio |
| Correo | X | | | | X |
| Password | X | | | | X |
| Mostrar Contrasena | | X | | | |
| Ingresar | | X | | | |
| Recordar | | X | | | |

| |
|---|
| Datos de Identificacion |
| Usuario |

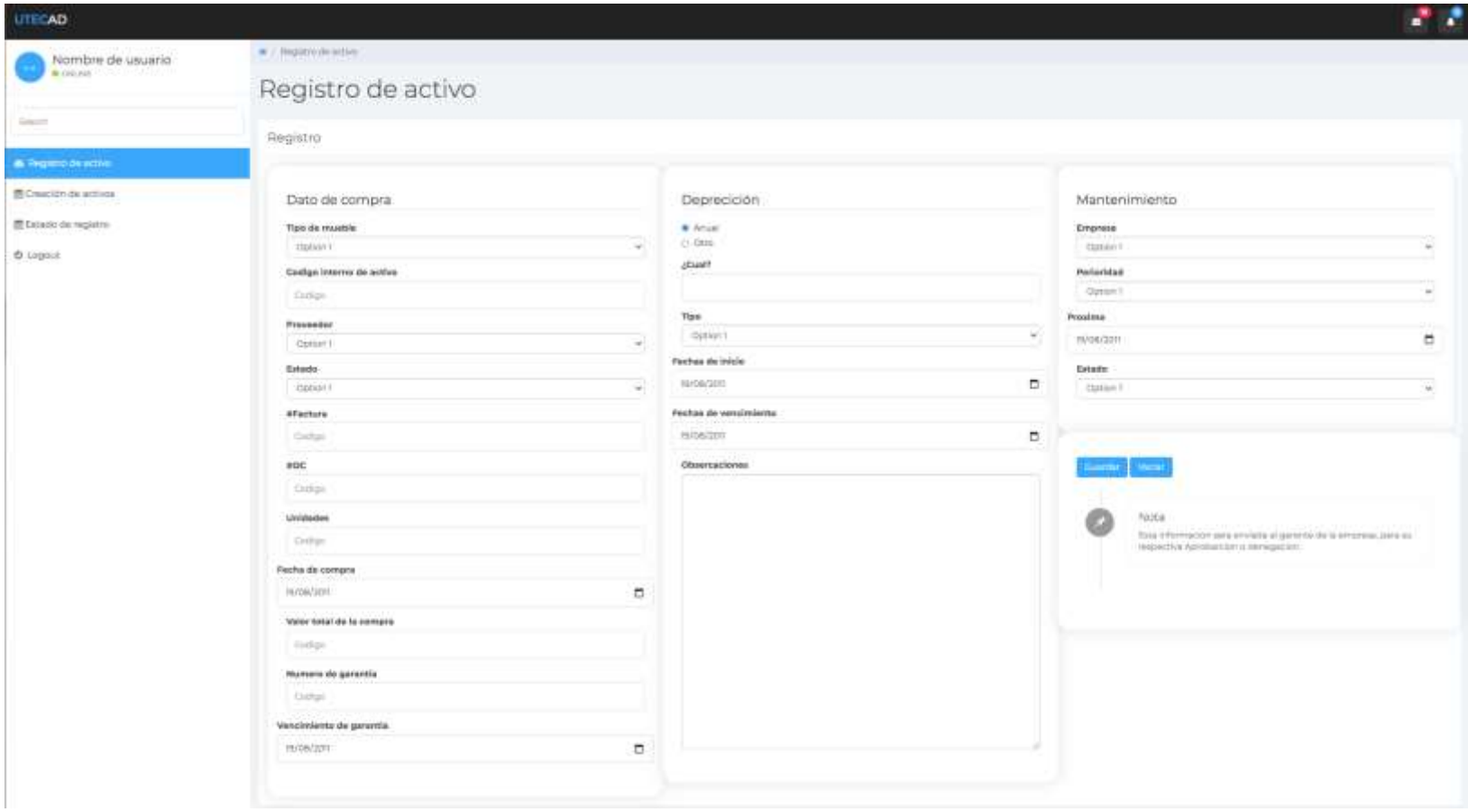



| | | | | | |
|---|---|---|---|---|---|
| Pantalla: | ESTADO DE REGISTRO | | | Interfaz | 2 |
| Descripcion: | Pantalla genérica para conocer el estado de una transacción y/o los datos resultantes al envío de una transacción. | | | | |
| Objetivo: | Proporcionar visualmente el estado de una transacción enviada par aaprobacion y po rconsiguiente el estado final de la misma al afectar los registros de la Base de Datos | | | | |
| Usuarios: | Aplica para todos los usuarios del sistema. | | | | |
| Datos de Entrada | | | | | |
| Detalle | Tipo | | | | |
| | Digitar | Seleccionar | Recuperar | Generar | Obligatorio |
| cerrar | X | | | | |
| Datos de Identificacion | | | | | |
| Usuario | | | | | |




| Pantalla: | CRAECOIN DE ACTIVOS | | | Interfaz | 3 |
|---|---|---|---|---|---|
| Descripción: | Creación de un activo en el sistema con todos los datos iniciales que requiere. | | | | |
| Objetivo: | Incorporar todos los datos concernientes al activo para ser incorporado al sistema | | | | |
| Usuarios: | Aplica para todos los usuarios del sistema. | | | | |
| Datos de Entrada | | | | | |
| Detalle | Tipo | | | | |
| | Digitar | Seleccionar | Recuperar | Generar | Obligatorio |
| Cajas de Texto | X | | | | |
| Guardar | | X | | | |
| Verificar | | X | | | |
| Modificar | | X | | | |
| Cerrar | | X | | | |
| Datos de Identificación | | | | | |

| |
|---|
| Usuario |




| | | | | | |
|---|---|---|---|---|---|
| Pantalla: | REGISTRO DE ACTIVO | | | Interfaz | 4 |
| Descripción: | Pantalla genérica encargada del registro de toda la información referente a un activo fijo, a su vez será enviada dentro de una transacción | | | | |
| Objetivo: | Contener dentro de una transacción las instrucciones necesarias para que la lógica del negocio sepa qué hacer con la información enviada. | | | | |
| Usuarios: | Aplica para todos los usuarios del sistema. | | | | |
| Datos de Entrada | | | | | |
| Detalle | Tipo | | | | |
| | Digitar | Seleccionar | Recuperar | Generar | Obligatorio |
| Cajas de Texto | X | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Cerrar | | X | | | |
| Cancelar | | X | | | |
| Enviar | | X | | | |
| Datos de Identificación | | | | | |
| Usuario | | | | | |

**REPORTES**

A continuación, el reporte genérico designado para el 100% de la repostería a utilizar dentro del Sistema de Activo Fijo. Dado que el sistema utiliza mensajes XML para la transaccionabilidad del mismo, se ha decidido crear un reporte genérico que será alimentado por datos XML en el motor de Apache FOP (https://xmlgraphics.apache.org/fop/).

**REPORTE: Reporte Generico**

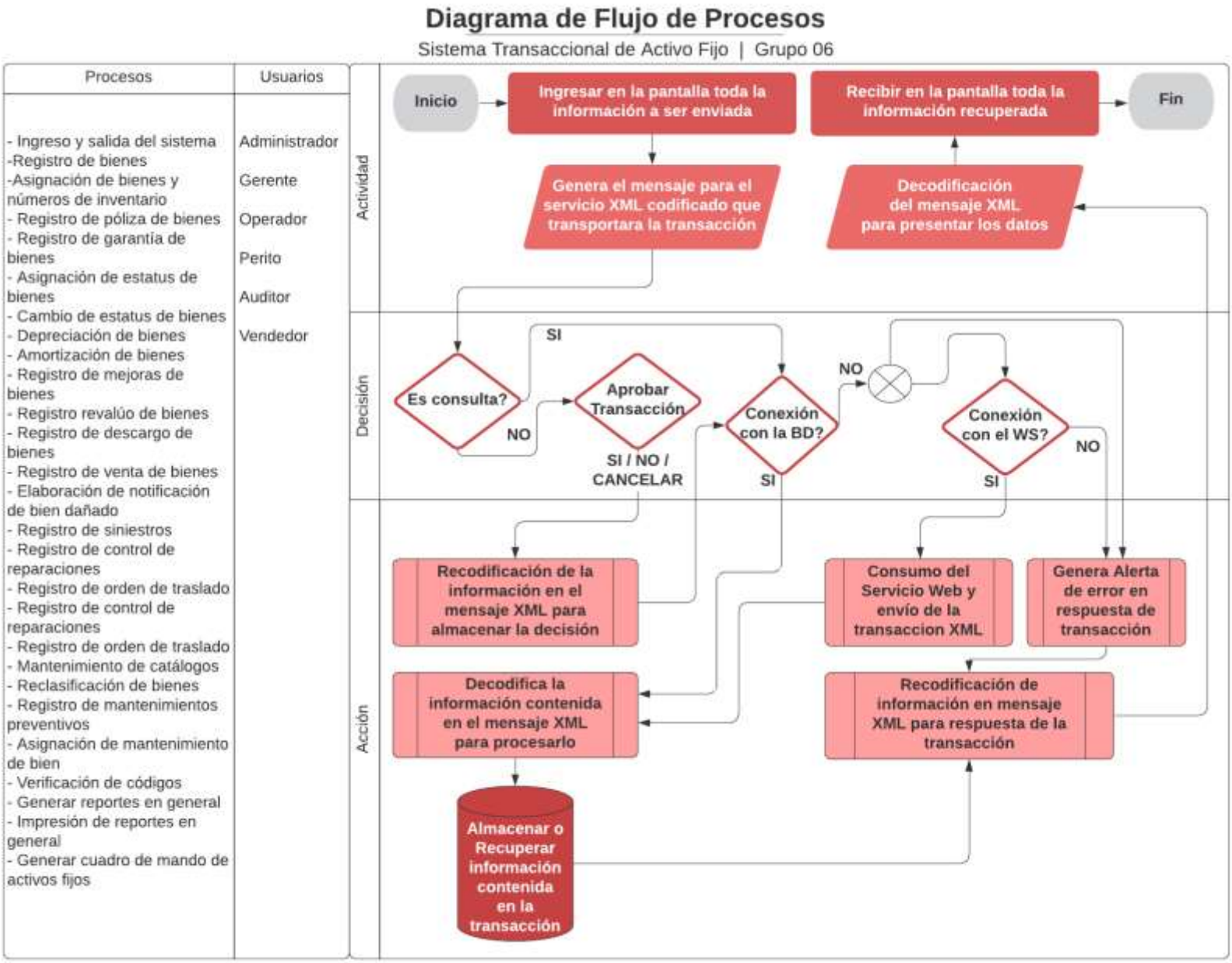
| | CAMPO | CAMPO | CAMPO | CAMPO | CAMPO |
|---|---|---|---|---|---|
| 1 | esta es una representacion de los datos | esta es una represent acion de los datos | 11 | esta es una representacion de los datos | 100000100001 |
| 2 | esta es una representacion de los datos | esta es una represent acion de los datos | 100 | esta es una representacion de los datos | El Salvador |
| 3 | esta es una representacion de los datos | esta es una represent acion de los datos | 100 | esta es una representacion de los datos | Salvadoreno |
| 4 | esta es una representacion de los datos | esta es una represent acion de los datos | 1 | esta es una representacion de los datos | 1 |
| 5 | esta es una representacion de los datos | esta es una represent acion de los datos | 11 | esta es una representacion de los datos | 200000100011 |
| 6 | esta es una representacion de los datos | esta es una represent acion de los datos | | esta es una representacion de los datos | 100000100001 |
| 7 | esta es una representacion de los datos | esta es una represent acion de los datos | 11 | esta es una representacion de los datos | 200000100011 |
| 8 | esta es una representacion de los datos | esta es una represent acion de los datos | | esta es una representacion de los datos | 2020/09/24 10:00:01 |

Fecha: Sept 24, 2020

## DISEÑO DE PROCESOS

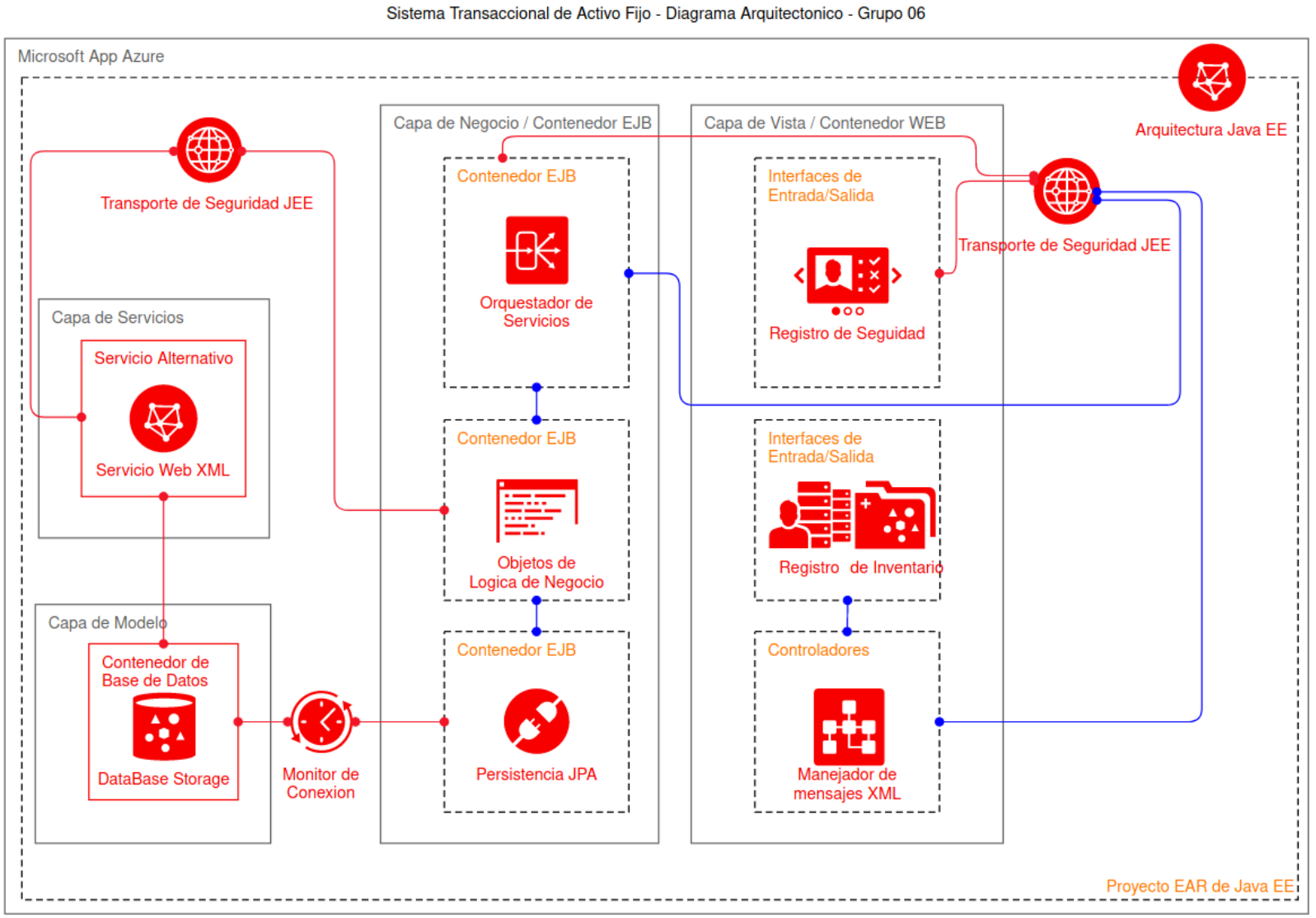
Debido al planteamiento de arquitectura JEE descentralizada que se ha diseñado, un Orquestador de Servicios se encargará del manejo de servicios de transacciones a lo largo del sistema con el apoyo del Transporte de Seguridad JEE, con ello todos los flujos de procesos se codificaran en mensajes XML para viajar y comportarse como transacciones manejadas por el Orquestador de Servicios. Por lo anterior, el mismo flujo aplica para todos los procesos e implica a todos los usuarios por igual.

## FLUJOS DE PROCESOS



## DISEÑO DE ARQUITECTURA

## DIAGRAMA ARQUITECTÓNICO



## DESCRIPCIÓN DE CADA COMPONENTE

### Microsoft App Azure

Servicio de hospedaje web totalmente administrado que permite crear aplicaciones web, back-ends móviles y API RESTful. Desde sitios web pequeños hasta aplicaciones web con una escala global, con las opciones de precios y rendimiento que se adaptan a necesidades varias.

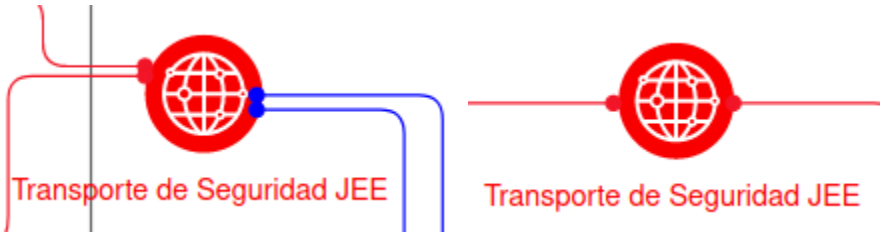
Orientada a crear, implementar y escalar rápidamente APIs y aplicaciones web al gusto y neceisdad, lista para trabajar con .NET, .NET Core, Node.js, Java, Python o PHP, bien en contenedores o en ejecución en Windows o Linux. Satisface requisitos exigentes de rendimiento empresarial, seguridad y cumplimiento normativo usando una plataforma de confianza totalmente administrada que puede controlar más de 40,000 millones de solicitudes al día.

### Arquitectura Java EE


Arquitectura Java EE (Edicion Empresarial) es una colección de especificaciones API diseñadas para trabajar juntas cuando se desarrollan aplicaciones Java empresariales del lado del servidor. Java EE es un estándar; existen múltiples implementaciones de las especificaciones Java EE y este hecho evita el bloqueo del proveedor, ya que el código desarrollado según la especificación Java EE se puede implementar en cualquier servidor de aplicaciones compatible con Java EE con modificaciones mínimas o sin modificaciones y se ha utilizado para desarrollar aplicaciones empresariales durante muchos años.

Proporciona una forma estándar de implementar muchos aspectos de las aplicaciones empresariales, como manejar solicitudes web, acceder a bases de datos, conectarse a otros sistemas empresariales e implementar servicios web. A lo largo de los años, ha evolucionado y ha hecho que el desarrollo de aplicaciones empresariales sea más fácil que antes. También cambió su nombre de J2EE a JEE después de la versión 1.4 de J2EE y, más recientemente, de Java Enterprise Edition a Jakarta Enterprise Edition en febrero de 2018. Actualmente, JEE está en la versión 8.

**Transporte de Seguridad JEE**




A menudo, la seguridad no se considera un elemento de diseño fundamental en un lenguaje de programación. En consecuencia, los desarrolladores de software incorporan mecanismos de seguridad al final de un proyecto a pesar de los recientes ataques de alto perfil. Sin embargo, Java utiliza una filosofía completamente diferente en la que la seguridad se considera un elemento de diseño crucial. Conoce muchas políticas y permisos de seguridad incluso antes de que se cargue la primera clase. Tenga en cuenta que solo los atacantes no insertan código malicioso; Las aplicaciones mal escritas pueden comprometer inadvertidamente los recursos a nivel del sistema o afectar el rendimiento de otras aplicaciones. Las aplicaciones Java no son víctimas de virus debido a la insistencia del lenguaje en una política firme y un diseño de permisos que con sus características principales y extensiones de seguridad, nos permite escribir potentes aplicaciones seguras. La seguridad incluye seguridad del idioma, criptografía, infraestructura de clave pública (PKI), autenticación, comunicación segura y control de acceso.

La tecnología de seguridad de Java abarca para el transporte de datos una amplia gama de áreas, incluida la criptografía, la infraestructura de clave pública, la comunicación segura, la autenticación, el control de acceso, la seguridad del idioma, un amplio conjunto de API, herramientas e implementaciones de algoritmos, mecanismos y protocolos de seguridad de uso común. Estos juntos proporcionan un marco de seguridad integral para desarrollar y administrar las aplicaciones, asi mismo se aplican los tres principios de diseño de las API de seguridad: independencia de la implementación, interoperabilidad de la implementación y extensibilidad del algoritmo.

**Capa Vista / Contenedor Web**

Formada por la lógica de aplicación, que prepara datos para su envío a la capa de negocio y procesa solicitudes a traves del manejador de mensajes XML desde el Orquestador de Servicios para su envío a la lógica de negocios. La lógica en esta capa está formada normalmente por J2EE como JSP y JSF que prepara los datos para enviarlos en formato XML, y que reciben solicitudes para procesarlas. Esta capa incluye el Registro de Inventario (o pantallas del sistema), el Manejador de Mensajes XML y el Registro de Seguridad que brinda acceso personalizado y seguro a los servicios del Orquestados en la capa de servicios de negocio.

### Capa de Negocio / Contenedor EJB

Consiste en la lógica que realiza las funciones principales de la aplicación: procesamiento de datos, implementación de funciones de negocios, coordinación de varios usuarios y administración de recursos externos como el acceso a bases de datos o sistemas heredados. Esta capa esta formada por componentes firmemente acoplados que se ajustan al modelo de componentes distribuidos de J2EE como objetos Java, componentes EJB o beans conducidos mediante mensajes. Montados en componentes J2EE individuales para ofrecer servicios de negocios complejos como el presente servicio de inventario. Los componentes individuales y el Orquestador de servicios se encapsulan como servicios que no estan del todo acoplados en un modelo de arquitectura orientada a servicios, que se ajuste a los estándares de la interfaz SOAP (Simple Object Access Protocol). Los servicios de negocios son contenedores de servidores independientes de mensajería o un servidor de calendario empresarial.

### Capa de Modelo

Formada por servicios que proporcionan los datos persistentes utilizados por la lógica de negocios. Los datos estan almacenados en un sistema de administración de bases de datos incluyendo información de recursos y directorios almacenada en un almacén de datos de protocolo ligero. Los servicios de datos incluyen alimentación de datos de orígenes externos mediante una capa de servicios web.

### Capa de Servicios

La clave de una arquitectura sólida es una capa de servicios bien definida. Esta capa expone la lógica de negocio a los clientes, cómo interfaces web o capas remotas. Esta capa consistirá en múltiples interfaces, cada uno con un contrato bien definido y siendo para uno de ello la funcion principal de atender al Registro de seguridad o autenticacion del usuario por medio de un Servicio Web, el resto de interfaces estaran definidas como un acceso secundario si la comunicación directa del servidor principal con la base de datos se pierde en algun momento.

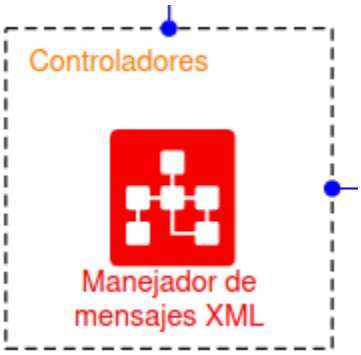
### Interfaces de Entrada / Salida - Registro de Seguridad

Permite acceder fácilmente a la autenticación de los usuarios y al mismo tiempo garantizar que se administren sus credenciales de inicio de sesión de forma segura por medio de un Servicio Web dedicado a esta tarea, lo que facilita el flujo de inicio de sesión para todos los usuarios del sistema y facilita el acceso a información básica de perfil sobre usuarios autenticados.

**Interfaces de Entrada / Salida - Registro de Inventario**


Comprende el escritorio de trabajo y las pantallas del sistema en las que se efectuan el registro de transacciones del sistema para trabajar con el inventario de activo fijo, asi como tambien comprende la aprobacion, negacion o cancelacion de estas transacciones en un entorno intuitivo que facilita la gestion de los datos.

**Interfaces de Entrada / Salida - Manejador de mensajes XML**




Es la parte del patron MVC correspondiente al controlador que gestiona la recoleccion de datos de las pantallas y las convierte en mensajes XML encriptados que se envian al Orquestador de Servicios por medio de Transporte de Seguridad con un encabezado que indica el servicio a utilizar esto a su vez es tratado por la logica de negocio del sistema. A su vez recibe mensajes XML encriptados con informacion o mensajeria de respuesta desde el Orquestador de Servicios para ser mostrados en pantalla o indicar a la sesion como tal, los cambios recuperados de los mensajes XML.

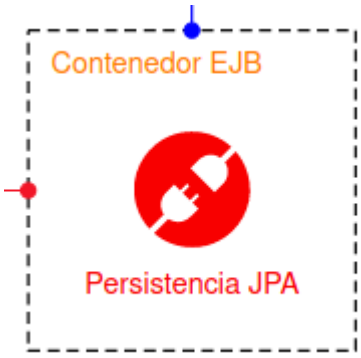
**Orquestador de Servicios**


Hace uso de un recurso central en el contenedor EJB que lleva el control de los servicios invocados en la realización de una tarea y coordina la ejecución de las diferentes operaciones de logica de negocios sobre dichos servicios. Los servicios orquestados no conocen que están implicados en un proceso de composición y que forman parte de un proceso de negocio de nivel más alto, solamente el Orquestador por medio de mensajes XML es consciente de esta centralizacion mediante llamadas explícitas operaciones XML (como servicios invocados) en el orden en el que se deben llamar para que sean utilizados en todos los procesos de negocio.

**Orquestador de Servicios**


Contiene las clases de definición para cada servicio del Orquestador con su logica de negocio para el tratamiento de los datos desde la vista hacia la base de datos y viceversa en cada servicio invocado.

**Persistencia JPA**




Es un marco ligero de Java Persistence API basado en POJO para la persistencia de los datos que ofrece soluciones a los desafíos arquitectónicos de integrar la persistencia del sistema.

**Monitor de Conexion**


Objeto dedicado a verificar la disponibilidad del enlace directo entre el servidor y la base de datos, ya que en caso de fallar, redirecciona el flujo de datos a la capa de servicios para consumir los servicios web ahi declarados.

**Contenedor de Base de Datos**


Sistema de Administracion de Base de Datos (DBMS) dedicado al almacenamiento de los datos enviados por el sistema, asi como el envio de los mismos ante peticiones autenticamente generadas.

**Servicio Alternativo**


Servicios Web dedicados a proveer data en modo contingencia cuando ha fallado la conexión directa del servidor con la base de datos. Su función principal es la de proveer un servicio de autenticacion segura para el Registro de Seguridad.

## DISEÑO DE SEGURIDAD

La seguridad física y logica, las tecnologías virtuales utilizadas para proteger un Data Center o solucion en la nube que alberga nuestro sistema transaccional (en adelante denominada “Solucion” o “Soluciones”) contra amenazas y ataques externos, contemplan una instalación que almacena infraestructura de TI, compuesta por computadoras en red y almacenamiento que se utiliza para organizar, procesar y almacenar grandes cantidades de datos. Aunque las Soluciones reducen el costo de implementar redes propias y servidores de computación centralizados, estas Soluciones brindan servicios tales como soluciones IaaS, SaaS, PaaS, DcaaS, almacenamiento de datos, respaldo y recuperación, administración de datos y redes. Debido a que contendran información confidencial de los datos del cliente y propiedad intelectual, los sitios tienen que estar protegidos digital y físicamente.

## SEGURIDAD FÍSICA

Las Soluciones son complejas y para protegerlas, los componentes de seguridad deben considerarse por separado, pero al mismo tiempo siguen una política de seguridad integral. La seguridad se puede dividir en física y software de seguridad. La seguridad física abarca una amplia gama de procesos y estrategias que se utilizan para prevenir interferencias externas. El software o la seguridad virtual impiden que delincuentes cibernéticos ingresen a la red al evitar el firewall, descifrar contraseñas o superar otras lagunas.

Respecto de la seguridad física: Las características de seguridad más obvias para nuestra Solucion están relacionadas con el diseño y la distribución. La Solucion en sí puede diseñarse como una unidad de propósito único o multipropósito, esta última funciona como un espacio compartido y puede albergar negocios no relacionados con el centro de datos. Una Solucion generalmente se construye lejos de las carreteras principales para establecer zonas de amortiguamiento formadas por una combinación de barreras de jardinería y a prueba de choques que a su vez pueden albergar el servicio en la nube que se necesita.

El acceso a las instalaciones de la Solucion es bastante limitado. La mayoría no tiene ventanas exteriores y relativamente pocos puntos de entrada. Los guardias de seguridad en el interior del edificio monitorean la actividad sospechosa utilizando imágenes de cámaras de vigilancia instaladas a lo largo del perímetro exterior. Los visitantes pueden usar la autenticación de dos factores para ingresar al edificio, incluido el escaneo de tarjetas de verificación de identidad personal (PIV) y el ingreso de un código de acceso personal. Los lectores de tarjetas de identificación de empleados y los sistemas biométricos, como los lectores de huellas dactilares, los escáneres de iris y el reconocimiento facial, también se pueden usar para permitir la entrada.

Respecto del Software de Seguridad: Los ataques de Hackers o Hackeo, el malware y el spyware son las amenazas obvias para los datos almacenados en una Solucion. Una herramienta de gestión de eventos e información de seguridad (SIEM) ofrece una vista en tiempo real de la postura de seguridad de una Solucion. Un SIEM ayuda a proporcionar visibilidad y control en todo, desde sistemas de acceso y alarma y sensores en la cerca perimetral de la Solucion.

La creación de zonas seguras en la red es una forma de colocar la seguridad en la Solucion. Los administradores pueden dividir las redes en tres zonas:

un área de prueba con una gran flexibilidad

zona de desarrollo con un entorno un poco más estricto

una zona de producción con solo equipos de producción aprobados.

Pero antes de que se implementen aplicaciones y código, ciertas herramientas pueden usarse para analizar las Soluciones en busca de vulnerabilidades que puedan ser fácilmente explotadas, y luego proporcionar métricas y capacidades de remediación. El código se puede ejecutar a través de un escáner para verificar si hay desbordamientos de búfer u otras vulnerabilidades. Con el aumento de la computación en la nube, la visibilidad de los flujos de datos es una necesidad, ya que podría haber malware escondido dentro del tráfico que de otra manera sería legítimo.

Las Soluciones se pueden colocar en cuatro niveles, cada nivel está asociado con una función comercial específica y establece un criterio apropiado para la refrigeración, el mantenimiento y la capacidad para resistir una falla. Básicamente, cada nivel muestra qué tan tolerante a las fallas es ese sistema, medido en tiempo de actividad, y qué tipo de seguridad puede necesitar:

Capa 1 + 2: Estos son generalmente se utilizarán por las sitios remotos que no proporcionan la entrega en tiempo real de productos o servicios. El nivel 1 comprende componentes de capacidad no redundantes, como un solo enlace ascendente y servidores. El nivel 2 incorpora los requisitos del nivel 1, pero agrega componentes de capacidad redundantes.

Capa 3 + 4: Rigurosos requisitos de tiempo de actividad y la viabilidad a largo plazo suelen ser la razón para seleccionar soluciones estratégicas que se encuentran en la infraestructura de sitios de nivel 3 y nivel 4. Estas Soluciones se consideran más robustos y menos propensos a fallas. El nivel 3 comprende los requisitos del nivel 1 + nivel 2, pero agrega equipos de doble alimentación y varios enlaces ascendentes. El nivel 4 comprende los requisitos de los tres niveles anteriores pero con componentes que son totalmente tolerantes a fallos, incluidos los enlaces ascendentes, el almacenamiento, los enfriadores, el HVAC y más.

**Respuesta a incidentes**

Como su nombre lo indica, la respuesta a incidentes es el conjunto de procesos y procedimientos que se inician una vez que se ha declarado un incidente de seguridad. En la computación moderna, los incidentes van desde un único punto final comprometido hasta compromisos completos de red que resultan en violaciones masivas de datos. Las violaciones de datos y los ataques en toda la empresa se están volviendo cada vez más comunes, por lo que la respuesta a incidentes ha crecido en significado más allá de estos procesos y procedimientos para abarcar una disciplina completa dentro de la seguridad de la información, los diversos procesos involucrados en la respuesta a incidentes, las herramientas y las

opciones tecnológicas, y las formas más comunes de análisis técnico que probablemente deba realizar durante un incidente.

**Procesos**

Los procesos de respuesta a incidentes son un componente integral de poder reaccionar rápidamente en caso de un incidente, determinar un no incidente, operar de manera eficiente durante un incidente y mejorar después de un incidente. Tener procesos implementados antes de que comience un incidente pagará dividendos a largo plazo.

**Procesos Pre-Incidentes**

Los procesos asociados con la respuesta a incidentes no están relacionados simplemente con lo que sucede durante un incidente. Si no hay procesos implementados para reconocer que se está produciendo un incidente, que se debe iniciar el proceso de respuesta al incidente y que se notifique a los responsables de la respuesta al incidente, no tiene sentido que haya procesos para tratar el incidente, ya que nunca habrán de ser llamados.

Los procesos previos al incidente no necesitan ser complejos; de hecho, definitivamente no deberían serlo, el objetivo de estos procesos Disaster Recoveryes simplemente determinar si hay un incidente potencial e iniciar el proceso de respuesta al incidente. Habiendo pasado por varias iteraciones de respuesta a incidentes internos, podemos decir que los procesos más efectivos con los que he trabajado incluyen los siguientes:

Aproveche los procesos existentes para tratar con eventos: la mayoría de las organizaciones se enfrentan a interrupciones, problemas de configuración, problemas informados por los usuarios y otros eventos. No intente configurar un conjunto paralelo de procesos, sino aprovechar lo que ya está allí, con toda probabilidad, las mismas personas que se ocupan de estos problemas serán las primeras en enterarse de un problema de todos modos. Simplemente modifique o complemente los procesos existentes para incluir llamar al contacto de respuesta a incidentes en caso de que se produzca un incidente esperado, de manera similar a como ya saben para llamar a la persona de Unix de guardia cuando un host de Linux falla en medio de la noche.

Defina un incidente: si no define lo que clasifica como un incidente, se le llamará por cada llamada de soporte o no se le llamará durante una violación de cuatro millones de registros. Si no es sencillo definir qué es un incidente, puede optar por palabras como "una vez que el

administrador haya determinado que un evento es un incidente de seguridad ..." De esta manera, al menos habrá definido que cualquier evento ya habrá progresado más allá de la clasificación por el soporte de primera línea y alguien con experiencia suficiente para tomar la decisión ha emitido un juicio.

El resultado de un proceso previo al incidente es casi siempre iniciar el proceso de IR al declarar un incidente y llamar al contacto para obtener una respuesta al incidente.

**Procesos de incidentes**

Los procesos que tienen lugar durante un incidente, particularmente desde una perspectiva tecnológica, no pueden ser demasiado prescriptivos. Los incidentes, al igual que muchos problemas operativos, son demasiado variados y numerosos para prescribir cursos de acción precisos para todas las eventualidades. Sin embargo, hay algunos procesos que vale la pena seguir:

Defina un administrador de incidentes: no tiene que ser la misma persona para cada incidente, sino que debe ser una persona lo suficientemente avanzada como para tomar decisiones y capacitar a otros para completar tareas. El administrador de incidentes ejecutará el esfuerzo de respuesta y tomará las decisiones.

Defina las comunicaciones internas: la comunicación entre todos los que trabajan en el incidente para evitar la duplicación de trabajos, promover el intercambio de información y garantizar que todos trabajen para lograr un objetivo común es la clave. Recomendaríamos:

Abra una "sala de guerra". Es decir, use una oficina o sala de reuniones para realizar el rol de centro de operaciones para cualquier persona en la misma ubicación física. Esto se utiliza como el punto central para la coordinación de esfuerzos.

Mantenga un puente de conferencia abierto en la sala de guerra. Esto permite que las personas que están alejadas de la ubicación física se registren, actualicen a aquellos en la sala de guerra y obtengan comentarios. Si no hay una sala de guerra física, esto a menudo servirá como una sala de guerra virtual.

Mantengase reuniones periódicas de actualización. Las actualizaciones periódicas permiten que las personas se alejen, trabajen de forma más concentrada, y que informen con regularidad en lugar de sentirse como si fueran revisadas y que informaran al azar. Por lo general, reunirse cada hora funciona bien hasta que la situación se comprende bien.

Asignar la tarea de comunicarse internamente a las partes interesadas. La gerencia generalmente querrá mantenerse al tanto de un incidente mayor. Sin embargo, la comunicación esporádica de varias personas puede enviar mensajes mixtos y ser frustrante tanto para la gerencia como para el equipo de respuesta a incidentes. Un único punto de comunicación entre los dos permite a las partes interesadas recibir actualizaciones frecuentes y medidas.

Defina las comunicaciones externas: en muchos casos, pero no en todas, es posible que se requiera alguna comunicación externa. Por lo general, esto se debe a que los clientes u otros departamentos se verán afectados por el incidente de alguna manera. Este tipo de comunicación no debe tomarse a la ligera, ya que afecta la imagen pública de la organización y el departamento de tecnología interna. Si está considerando realizar alguna comunicación externa usted mismo, en lugar de permitir que su comunicación corporativa o su equipo de relaciones públicas lo hagan, le sugerimos que lea la publicación del blog "Crisis Comms for IR" de Scott Roberts.

Determine los objetivos clave: Al determinar los objetivos que desea alcanzar en caso de un incidente, puede asegurarse de que todas las acciones se tomen teniendo en cuenta estos objetivos. Por objetivos no nos referimos simplemente a "arreglarlo", sino a consideraciones tales como "preservar la cadena de custodia como evidencia" o "minimizar el tiempo de inactividad".

Procesos tecnológicos de alto nivel: como se mencionó anteriormente, es difícil dar cuenta de todas las eventualidades, por lo que ser prescriptivo con recursos basados en tecnología puede ser difícil; sin embargo, hay algunos procesos de alto nivel que pueden implementarse. Por ejemplo, puede haber políticas con respecto a la toma de instantáneas de los sistemas afectados para preservar la evidencia, asegurar que el personal deje de iniciar sesión en los sistemas afectados o un apagón en la discusión de incidentes por correo electrónico en caso de que un atacante esté leyendo un correo electrónico interno y reciba una alerta.

lan para el largo plazo: muchos incidentes terminan en unas pocas horas, pero muchos duran sustancialmente más, a menudo semanas. Es tentador extraer todos los recursos para ayudar en un incidente con la esperanza de llegar a una conclusión oportuna, pero si queda claro que este no es el caso, debe prepararse para un curso de acción a más largo plazo. Asegúrese de que las personas sean enviadas a descansar para que puedan entrar y cubrir el próximo turno,

y mantener alimentadas y regadas a las personas que trabajan para prevenir la fatiga. Intenta no quemar a todos, ya que este puede ser un juego de resistencia.

**Procesos posteriores al incidente**

Una vez que termina un incidente, es muy valioso realizar una sesión de lecciones aprendidas. Esto permite comentarios sobre lo que funcionó bien y lo que funcionó menos bien. También le permite la oportunidad de actualizar los procesos, determinar los requisitos de capacitación, cambiar la infraestructura y, en general, mejorar en función de lo que aprendió del incidente.

Se recomienda que esta sesión se lleve a cabo poco tiempo después del cierre del incidente. Esto ofrece algunos días para que las personas reflexionen sobre lo sucedido, adquieran cierta perspectiva y se recuperen, sin dejarlo tanto tiempo que los recuerdos se desvanezcan o se distorsionen con el tiempo. El uso de esta sesión para actualizar la documentación, las políticas, los procedimientos y los estándares también permitirá la actualización de mesas y simulacros.

**Herramientas y tecnologia**

Sería fácil enumerar una gran cantidad de tecnologías que suelen utilizar los profesionales de respuesta a incidentes, especialmente en el campo de la ciencia forense digital. Sin embargo, la falta de experiencia en esta área puede facilitar la mala interpretación de los resultados, ya sea a través de la falta de experiencia con las herramientas específicas o al no comprender el contexto de lo que se entiende completamente.

Comprender completamente un entorno, saber lo que significan los distintos registros, saber qué debería y no debería estar presente, y aprender a usar las herramientas que ya están presentes puede aumentar enormemente las posibilidades de gestionar un incidente en curso. La mitad del incidente no es el momento de aprender a realizar una investigación forense; es mejor dejarlo a alguien que tenga alguna experiencia previa en este campo. Dicho esto, se puede lograr una apreciación de alto nivel de lo que puede suceder durante un incidente revisando algunos temas de alto nivel. También analizamos algunas herramientas de ejemplo que se pueden usar para evaluar lo que sucede en un entorno durante un incidente.

**Análisis de registro**

El primer puerto de todo, como con cualquier tipo de problema operacional, es, por supuesto, el humilde archivo de registro. Los archivos de registro de la aplicación y del sistema operativo pueden contener una gran cantidad de información y proporcionar punteros valiosos a lo que ha sucedido. Si los registros se almacenan en el host que los generó, debe tener en cuenta el hecho de que si alguien compromete a ese host, puede modificarlos fácilmente para eliminar la evidencia de lo que está sucediendo. Si es posible, se deben consultar los registros almacenados en su plataforma de Gestión de información y eventos de seguridad (SIEM), en lugar de consultar los registros en el dispositivo de destino. Esto no solo reduce las posibilidades de alteración de registros, sino que también brinda a las instalaciones la capacidad de consultar registros de todo el estado a la vez, lo que permite una visión más integral de la situación. Un SIEM también tiene la capacidad de mostrar si se ha producido un espacio en los registros.

Al revisar los registros en un SIEM, es probable que sea necesario utilizar las propias herramientas de consulta de registro y el idioma de búsqueda del SIEM. También es posible que el uso de comandos como curl o scripts personalizados accedan a los datos a través de una API. Si no se puede acceder a los registros en un SIEM, se recomienda tomar una copia, si es posible, y analizarlos localmente con las herramientas preferidas. Personalmente, optamos por una combinación de herramientas tradicionales de línea de comandos de Unix como grep, awk, sed y cut, junto con scripts escritos para casos de uso específicos.

**Análisis de disco y archivo**

El análisis de artefactos en los dispositivos de almacenamiento también puede proporcionar pistas sobre lo que sucedió durante un incidente. Por lo general, una imagen de disco proporcionará más información que solo examinar archivos, ya que no solo contiene los archivos almacenados en el disco que son visibles inmediatamente, sino también fragmentos potenciales de archivos eliminados que permanecen en el disco, fragmentos de datos que quedan en el espacio vacío. y archivos que han sido ocultados a través de kits de raíz. El uso de una imagen de disco también garantiza que no modifique accidentalmente el disco original, lo que garantiza la integridad del original en caso de que se produzcan procedimientos legales de algún tipo. Para obtener una copia de la imagen del disco, tradicionalmente significa eliminar un host y usar una herramienta como ddfldd o un equivalente comercial para tomar una imagen

del disco, que se guarda en otra unidad y luego se examina sin conexión. Desafortunadamente, esto causa tiempo de inactividad.

Una vez que se ha obtenido una imagen de disco, se pueden usar varias herramientas comerciales para analizar el sistema de archivos para descubrir archivos de interés, construir líneas de tiempo de eventos y otras tareas relacionadas. En la fuente abierta / espacio libre, los clásicos clásicos The Sleuth Kit y Autopsy siguen siendo los favoritos. Si todo lo que se desea es una simple recuperación de archivos, PhotoRec es una herramienta fácil de usar que produce resultados sorprendentemente buenos. A pesar del nombre, no se limita a las fotos.

**Análisis de memoria**

El código que se está ejecutando, incluido el código malicioso, reside en la RAM. Si puede obtener un volcado de memoria de un host comprometido, es decir, un archivo que contiene una copia byte-by-byte de la RAM, entonces se puede realizar un análisis para descubrir códigos maliciosos, enlaces de memoria y otros indicadores de lo que sucedió. . La herramienta más popular para analizar estos volcados de memoria RAM es el Marco de volatilidad (consulte la wiki en GitHub).

La obtención de volcados de RAM variará de un sistema operativo a otro y es un campo en constante cambio, por lo que recomendamos consultar la documentación de Volatility para conocer el método preferido más reciente. Sin embargo, para plataformas virtualizadas, no es necesario volcar la RAM utilizando el sistema operativo, ya que el host puede tomar una imagen de la memoria virtual. Los siguientes son los tres ejemplos más comunes de cómo lograr esto:

QEMU

```
pmemsave 0 0x20000000 /tmp/dumpfile
```

Xen

```
sudo xm dump-core -L /tmp/dump-core-6 6
```

VMWare ESX

```
vim-cmd vmsvc/getallvms
```

```
vim-cmd vmsvc/get.summary vmid
```

vim-cmd vmsvc/snapshot.create vmid [Name] [Description]

[includeMemory (1)] [quiesced]

**Análisis PCAP**

Si tiene alguna herramienta que detecte tráfico de red en línea o mediante un puerto de expansión o utilidades en línea como IDS / IPS o dispositivo de monitoreo de red, existe la posibilidad de que tenga archivos de ejemplo de captura de paquetes (PCAP). Los archivos PCAP contienen copias de los datos tal como aparecieron en la red y permiten que un analista intente reconstruir lo que estaba sucediendo en la red en un momento determinado.

Se puede utilizar una gran cantidad de herramientas para realizar el análisis PCAP; sin embargo, para un primer paso para comprender lo que está contenido en el tráfico, le recomendamos que utilice herramientas similares a IDS, como Snort o Bro Security Monitor, configuradas para leer desde un PCAP, en lugar de una interfaz de red en vivo. Esto capturará el tráfico obvio que desencadena sus firmas predefinidas. Algunas grapas para realizar el análisis de PCAP incluyen las siguientes herramientas:

tcpdump produce información de encabezado y resumen, volcados hexadecimales y volcados ASCII de paquetes que se pueden detectar en el cable o leer de archivos PCAP. Debido a que tcpdump es la línea de comandos, se puede usar con otras herramientas como sed y grep para determinar rápidamente las direcciones IP, puertos y otros detalles que ocurren con frecuencia para detectar tráfico anormal. tcpdump también es útil porque puede aplicar filtros a archivos PCAP y guardar la salida filtrada. Estos archivos de salida son, en sí mismos, PCAP más pequeños que pueden incorporarse a otras herramientas que no manejan PCAP grandes con tanta gracia como lo hace tcpdump.

Wireshark es la herramienta de facto para el análisis de datos PCAP. Proporciona una GUI completa que le permite al usuario realizar funciones como filtrar y rastrear una sola conexión, proporcionar análisis de protocolo y graficar ciertas características del tráfico de red observado. Sin embargo, Wireshark no maneja muy bien los archivos grandes, por lo que se recomienda prefiltrar con tcpdump.

tshark (incluido con Wireshark) es una versión de línea de comandos de Wireshark. No es tan intuitivo o fácil de usar, pero estar en la línea de comandos permite que se use junto con otras herramientas como grep, awk y sed para realizar un análisis rápido.

**Recuperación de desastres**

Los términos recuperación de desastres (DR) y planificación de la continuidad del negocio (BCP) a menudo se confunden y se tratan como intercambiables. Sin embargo, son dos términos diferentes pero relacionados. La continuidad del negocio se refiere a la continuación general del negocio a través de una serie de contingencias y planes alternativos. Estos planes pueden ejecutarse en función de la situación actual y las tolerancias del negocio para las interrupciones y demás. La recuperación ante desastres es el conjunto de procesos y procedimientos que se utilizan para alcanzar los objetivos del Plan de Continuidad del Negocio. Normalmente, BCP se extiende a toda la empresa, no solo a TI, incluidas áreas como oficinas secundarias y sistemas bancarios alternativos, energía y servicios públicos. DR suele estar más centrado en la TI y analiza tecnologías como las copias de seguridad y los recursos en caliente.

¿Por qué estamos hablando de DR y BCP en un libro de seguridad? La tríada de la CIA (confidencialidad, integridad y disponibilidad) se considera clave para casi todos los aspectos de la seguridad de la información, y BCP y DR se centran en gran medida en la disponibilidad, al tiempo que mantienen la confidencialidad y la integridad. Por esta razón, los departamentos de seguridad de la información a menudo están muy involucrados en las etapas de planificación de BCP y DR. En este capítulo, discutiremos el establecimiento de nuestros criterios objetivos, las estrategias para lograr esos objetivos y las consideraciones de prueba, recuperación y seguridad.

**Establecer objetivos**

Los objetivos le permiten asegurarse de que está cumpliendo de manera mensurable con los requisitos del negocio al crear una estrategia de DR y le permite tomar decisiones más fácilmente en relación con el equilibrio entre el tiempo y las consideraciones presupuestarias en comparación con el tiempo de actividad y los tiempos de recuperación.

**Objetivo Punto de Recuperación**

El objetivo de punto de recuperación (RPO) es el momento en el que desea recuperar. Es decir, determinar si necesita poder recuperar los datos hasta segundos antes de que ocurra el desastre, o si la noche anterior es aceptable, o la semana anterior, por ejemplo. Esto no tiene en cuenta el tiempo que tarda en realizar esta recuperación, solo el momento en el que se reanudará una vez que se haya

realizado la recuperación. Hay una tendencia a saltar directamente a segundos antes del incidente; sin embargo, cuanto más corto sea el RPO, más serán los costos y la complejidad invariablemente hacia arriba.

**Objetivo de tiempo de recuperación**

El objetivo de tiempo de recuperación (RTO) es el tiempo que tarda en recuperarse, independientemente del RPO. Es decir, después del desastre, cuánto tiempo hasta que se haya recuperado hasta el punto determinado por el RPO. Para ilustrar con un ejemplo, si opera un servidor que aloja su sitio web de folletos, el objetivo principal probablemente será que el servidor regrese rápidamente al uso operativo. Si el contenido tiene un día de antigüedad, probablemente no sea tan problemático como si el sistema tuviera transacciones financieras para las cuales la disponibilidad de transacciones recientes es importante. En este caso, una interrupción de una hora puede ser tolerable, con datos que no tengan más de un día de recuperación.

En este ejemplo, el RPO sería un día y el RTO sería una hora. A menudo, existe la tentación de que alguien de un departamento de tecnología establezca estos tiempos; sin embargo, debe ser impulsado por los dueños de negocios de sistemas. Esto es por múltiples razones:

A menudo es difícil justificar el costo de las soluciones de DR. Permitir que la empresa establezca requisitos y, posiblemente, restablecer los requisitos si los costos son demasiado altos, no solo permite tomar decisiones informadas con respecto a los objetivos, sino que también reduce las posibilidades de expectativas poco realistas sobre los tiempos de recuperación.

El personal de TI puede entender las tecnologías involucradas, pero no siempre tienen la perspectiva correcta para determinar cuáles son las prioridades de la empresa en tal situación.

La participación del negocio en los planes de DR y BCP facilita el proceso de discutir el presupuesto y las expectativas para estas soluciones.

**Estrategias de recuperación**

Se pueden implementar varias estrategias diferentes para satisfacer las necesidades de DR de su organización. Lo que sea más apropiado dependerá del RTO definido, RPO y, como siempre, del costo.

**Copias de seguridad**

La estrategia más obvia para recuperarse de un desastre es realizar copias de seguridad periódicas de todos los sistemas y restaurarlas a nuevos equipos. El nuevo equipo debe mantenerse en una instalación de recuperación de desastres u oficina secundaria dedicada, ubicada en algún lugar donde esté disponible la conectividad adecuada y los servidores puedan comenzar a operar de inmediato.

Históricamente, las copias de seguridad se realizaban a menudo en un medio basado en cinta, como las unidades DLT, que se enviaban físicamente a otra ubicación. Sin embargo, en los últimos tiempos, el costo del almacenamiento y la conectividad de la red se han reducido, por lo que las copias de seguridad a menudo se pueden realizar en un medio más confiable y fácilmente disponible, como un archivo en un disco duro remoto.

Las copias de seguridad generalmente tendrán un RPO más largo que otras estrategias, y las copias de seguridad no son continuas, sino que se ejecutan por lotes durante la noche, y no necesariamente todas las noches. El RPO será, en el mejor de los casos, el momento de la copia de seguridad más reciente. Además, las copias de seguridad fallan con frecuencia, por lo que el RPO es en realidad el momento de su copia de seguridad más reciente. El RTO variará según la velocidad de los medios de respaldo y la ubicación de los medios de respaldo en relación con el equipo de respaldo. Por ejemplo, si los medios de copia de seguridad deben enviarse físicamente a una ubicación, esto debe tenerse en cuenta.

**Espera “en caliente”**

Un modo de espera cálido es una infraestructura secundaria, idealmente idéntica a la primaria, que se mantiene en una sincronización aproximada con la infraestructura primaria. Esta infraestructura debe mantenerse a una distancia geográfica razonable de la primaria en caso de eventos como terremotos e inundaciones. En el caso de un desastre, los servicios se "cortarían" manualmente a la infraestructura secundaria. El método para hacerlo varía, pero a menudo está reenlazando las entradas de DNS de primario a secundario o modificando las tablas de enrutamiento para enviar tráfico a la infraestructura secundaria.

La infraestructura secundaria se mantiene sincronizada a través de una combinación de garantizar que los cambios de configuración y los parches se apliquen tanto a los procesos primarios como a los secundarios, y los procesos automatizados para mantener los archivos sincronizados. Idealmente, la configuración y la aplicación de parches se realizarán de manera automatizada utilizando un software de administración, sin embargo, este no suele ser el caso y puede causar problemas en caso de que existan diferencias.

El RPO es bastante corto en un modo de espera cálido, por lo general, independientemente de la frecuencia de los procesos de sincronización del sistema de archivos. El RTO es, sin embargo, largo que lleva el mecanismo de corte. Por ejemplo, con un cambio de DNS, esta es la cantidad de tiempo para realizar el cambio y los registros antiguos caducan en cachés para que los hosts usen el nuevo sistema. Con un cambio de enrutamiento, el RTO es, al menos, sin importar el tiempo que tome el cambio de enrutamiento y, si se usan protocolos de enrutamiento dinámico, se producirá la

convergencia de la tabla de enrutamiento. Sin embargo, este sistema se basa en tener una segunda infraestructura completa que no hace nada hasta que se produce un desastre.

**Alta disponibilidad**

Un sistema de alta disponibilidad suele ser un modelo como un clúster distribuido. Es decir, varios dispositivos en ubicaciones distribuidas, que comparten la carga durante los períodos de producción normales. Durante un desastre, uno o más dispositivos se eliminarán del grupo y los dispositivos restantes continuarán funcionando de manera normal. Además, continuarán procesando su parte de la carga adicional desde el dispositivo que ya no está operativo. Debido a la naturaleza de la alta disponibilidad, es típico que todos los dispositivos en el clúster estén completamente sincronizados, o muy cerca de ellos, y por esta razón, el RPO será muy corto.

Muchas tecnologías de agrupación en clústeres permiten que los dispositivos salgan del clúster y los otros dispositivos se ajustarán y compensarán automáticamente. Por esta razón, el RTO también puede ser más bajo que muchas otras soluciones. Si bien el RPO y el RTO son ventajosos cuando se usa un sistema de alta disponibilidad, no es gratis. El clúster debe tener suficiente capacidad para compensar el manejo de la carga adicional por cada nodo restante. En el caso de un desastre, esto significa ejecutar hardware que no se utiliza completamente para tener capacidad de reserva. Además, se requerirá una inversión adicional en áreas como el ancho de banda entre sitios. Mantener todos los dispositivos sincronizados para ejecutar una solución agrupada requiere un ancho de banda suficiente con una latencia lo suficientemente baja, lo que coloca requisitos adicionales en la infraestructura.

**Sistema alternativo**

En algunos casos, usar un sistema alternativo es preferible a ejecutar una copia de seguridad o un sistema secundario en el sentido tradicional. Por ejemplo, en el caso de que una solución de Voz sobre IP interna no esté disponible debido a un desastre, es posible que el plan no sea intentar volver a crear una instancia de la infraestructura de VoIP, sino simplemente cambiar al uso de teléfonos celulares hasta que en algun momento el desastre termine.

Esta estrategia no siempre tiene un RPO per se, ya que la recuperación del sistema existente no es parte del plan. Esta es la razón por la que este tipo de enfoque generalmente solo se realiza con sistemas que no contienen datos, sino que brindan un servicio, como los teléfonos. Sin embargo, existe un RTO medible en términos de la cantidad de tiempo que se tarda en cambiar a un sistema alternativo.

**Reasignación de funciones del sistema**

Un enfoque que puede resultar rentable es la reasignación de funciones del sistema, que es un híbrido de otras soluciones. Esta es la reutilización de sistemas no críticos para reemplazar sistemas críticos

en caso de una situación de desastre. No es aplicable a todos los entornos, por lo que debe considerarse cuidadosamente antes de ser utilizado como una estrategia.

Por ejemplo, si ya ejecuta dos centros de datos, estructure sus entornos de modo que para cualquier entorno de producción alojado en un centro de datos, su entorno de prueba, preproducción o control de calidad se aloje en el otro centro de datos. En este escenario, puede tener un sitio cercano a la producción listo, pero no inactivo, en todo momento. En el caso de un desastre, el entorno en cuestión dejará de funcionar como, por ejemplo, preproducción, y será promovido a un entorno de producción. Este enfoque requiere que los dos entornos estén lo suficientemente separados para que un desastre que afecte a uno no afecte al otro. El estado de los otros entornos debe controlarse estrechamente para que las diferencias de producción se conozcan y se cambien fácilmente para que coincidan con el estado de producción antes de comenzar a funcionar.

**Dependencias**

Una parte importante del desarrollo de una estrategia para DR y BCP es comprender las dependencias de todos los sistemas. Por ejemplo, si puede abrir con éxito un servidor de archivos en otra ubicación, no importa si el personal puede conectarse a él. Los servidores generalmente necesitan una conexión de red, el enrutamiento asociado, las entradas de DNS y el acceso a los servicios de autenticación, como Active Directory o LDAP. Si no se determinan las dependencias requeridas para cualquier sistema en particular, puede faltar el RTO para ese servicio.

Por ejemplo, si tiene un servidor de correo electrónico con un RTO de 1 hora y, sin embargo, la red de la que depende tiene un RTO de 3 horas, independientemente de la rapidez con la que el servidor de correo electrónico está funcionando, es posible que no se reanude la operación en Cualquier sentido significativo hasta que hayan transcurrido 3 horas. Al mapear las dependencias como esta, es mucho más fácil identificar RTO no realistas, o RTO de otros sistemas o servicios que necesitan ser mejorados para cumplir con estos objetivos. Caminar a través de mesas y ejercicios como se menciona en el Capítulo 1 ayudará a descubrir estas dependencias.

**Escenarios**

Al desarrollar planes de desastre potenciales, a menudo es útil repasar algunos escenarios de alto nivel y comprender cómo afectan su plan propuesto. Este ejercicio normalmente funciona de manera más efectiva con representantes de otros equipos de TI que pueden ayudarlo a analizar las implicaciones y dependencias de varias decisiones. Es útil considerar algunas categorías amplias de escenarios, aunque las que elija utilizar dependerán probablemente de sus propias circunstancias:

Falla de hardware de la plataforma de misión crítica: algo que está aislado en una sola plataforma, pero que es lo suficientemente importante como para causar un incidente de DR; por ejemplo, la falla del hardware del servidor para el entorno de producción de un sistema clave.

Pérdida de un centro de datos, potencialmente temporal, como durante un apagón, o quizás por períodos más prolongados, como un incendio o un terremoto.

Pandemia: en caso de una pandemia, los servicios pueden permanecer disponibles, pero el acceso físico puede no ser posible, lo que a su vez podría impedir que se realicen ciertos procesos, como el cambio físico de las cintas de respaldo, los usuarios que trabajan desde sus hogares causan una carga adicional de VPN. u otros servicios de acceso remoto.

**Invocando un fail over ... y back.**

Está muy bien tener un conjunto de planes de contingencia en el lugar y tener tiempos objetivo para alcanzarlos. Si no sabe cuándo se encuentra en una situación de desastre, los planes tienen poco sentido. Debe existir un proceso para determinar qué es y qué no es un desastre, y cuándo invocar el plan.

Puede haber algunos escenarios clave de alto nivel en los que el plan obviamente se pondrá en práctica. Por ejemplo, el evento del centro de datos en llamas suele ser suficiente para invocar la conmutación por error a los sistemas de copia de seguridad. Sin embargo, se debe tener cuidado de no ser demasiado prescriptivo o, de lo contrario, el riesgo de desviaciones menores de las situaciones descritas puede provocar que no se invoque el plan. De manera similar, no ser lo suficientemente descriptivo podría hacer que un administrador sin experiencia invoque un plan de DR sin necesidad. En este caso, ¿cómo determina cuándo invocar el plan? Una de las rutas más efectivas es tener una lista de personas o roles nombrados que estén autorizados para determinar cuándo la organización se encuentra en una situación de desastre y que el plan debe ejecutarse. El proceso para cualquier persona que no esté autorizada para tomar esta determinación es escalar a alguien que pueda, quien a su vez tomará la decisión. De esta manera, cualquier persona puede activar la alarma, pero la decisión final de ejecutar se deja en manos de una persona mayor y responsable.

Una de las áreas de DR y BCP que se pasan por alto a menudo es que, además de fallar a los sistemas de contingencia, tendrá que haber un proceso de cambio de nuevo después de que el desastre haya terminado. A diferencia del procedimiento de conmutación por error inicial, existe la ventaja de poder programar el cambio y tomar el tiempo adecuado para hacerlo. Sin embargo, este debe ser un proceso cuidadosamente planificado y ejecutado que es invocado una vez más por una persona autorizada. Siempre recuerde incluir la comunicación adecuada durante los posibles cortes, ya que puede ser un

momento de gran estrés. Un tiempo de inactividad nunca es demasiado grande para que ocurra una comunicación adecuada.

**Pruebas**

La recuperación de desastres puede ser extremadamente compleja, ya que muchas de las complejidades e interdependencias no son del todo obvias hasta que se encuentre en una situación de desastre. A veces encontrará que para completar la tarea, se requiere un archivo de un servidor que se encuentra actualmente bajo varios pies de agua. Por esta razón, es aconsejable, y bajo algunos regímenes de cumplimiento obligatorios, que se realicen pruebas de DR regulares. Por supuesto, nadie sugiere que el centro de datos se incendie y se intente recuperar. Elija un escenario y haga que los sistemas de reemplazo aparezcan dentro del RTO y RPO asignados. Esto debe completarse sin acceso a ningún sistema o servicio ubicado en la infraestructura afectada por el escenario que ha elegido. Se debe observar la prueba y tomar notas sobre lo que funcionó bien y lo que no. Realizar un informe posterior a la prueba con las personas clave involucradas, incluso si la prueba cumplió con todos los objetivos, es un proceso valioso que puede producir resultados muy útiles en la medida en que se aprende qué se puede mejorar en la preparación para la próxima vez. Los hallazgos del informe deben ser minuciosos con elementos de acción claros para los individuos con el fin de mejorar los planes y trabajar hacia un proceso más eficiente y sin problemas.

**Consideraciones de Seguridad**

Como con cualquier proceso, hay consideraciones de seguridad relacionadas con la mayoría de los planes. Estos se pueden resumir en algunas categorías clave:

Datos en reposo: muchos planes de contingencia requieren que los datos de los sistemas de producción se dupliquen y retengan en otro sitio. Esto se aplica tanto a los recursos en caliente como a las copias de seguridad tradicionales, por ejemplo. Siempre se debe recordar que estos datos tendrán controles en producción en línea con su valor y sensibilidad hacia la organización. Por ejemplo, puede estar encriptado, requerir autenticación de dos factores para acceder o estar restringido a un pequeño grupo de personas. Si no se aplican restricciones iguales a los sistemas de contingencia, los controles de acceso originales son en gran medida inútiles. Después de todo, ¿por qué un atacante se molestaría en intentar vencer la autenticación o el cifrado de dos factores en un sistema de producción cuando simplemente puede acceder a una copia relativamente desprotegida de los mismos datos desde un sistema de respaldo?

Datos en tránsito: para replicar los datos en un sistema secundario, probablemente tendrá que transmitirse a través de una red. Los datos transmitidos con el fin de recuperarse o prepararse para un

desastre deben tratarse tan cuidadosamente como en cualquier otro momento en que se transmiten los datos. La autenticación y el cifrado adecuados de los datos en la red aún deben aplicarse

Administración de parches y configuración: a menudo es fácil caer en la trampa de los sistemas de respaldo que no se mantienen en línea con el entorno de producción. Esto corre el riesgo de dejar equipos o vulnerabilidades mal parcheadas en su entorno para que un atacante las aproveche. En el caso de un desastre, estas vulnerabilidades podrían estar presentes en lo que se ha convertido en su sistema de producción. Aparte de los problemas de seguridad, no puede estar seguro de que los sistemas con diferentes configuraciones o niveles de parches funcionarán de la misma manera que sus contrapartes de producción.

Acceso del usuario: durante una situación de desastre, a menudo hay una sensación de "manos a las bombas" con el fin de garantizar que los entornos de producción sean capaces de funcionar tan pronto como sea posible. Se debe tener en cuenta que nadie puede acceder a todos los datos, en particular si los datos están sujetos a un régimen de cumplimiento normativo, como los que protegen los datos de salud o financieros personalmente identificables. Cualquier plan debe incluir el manejo continuo de este tipo de datos en línea con los procesos y procedimientos establecidos.

Seguridad física: A menudo, el sitio secundario puede no ser físicamente idéntico al sitio primario. Tomemos, por ejemplo, una empresa para la cual el entorno de producción principal se encuentra en un centro de datos administrado de instalaciones de terceros seguros, y la ubicación del desastre hace uso del espacio de oficina no utilizado en la sede. Un estándar más bajo de control de acceso físico podría poner a los datos o sistemas en riesgo si un atacante estuviera dispuesto a intentar ingresar físicamente a un edificio por la fuerza, el subterfugio o el sigilo.

## SEGURIDAD LÓGICA

Los ataques modernos pueden ocurrir por muchas motivaciones diferentes y son perpetrados por personas que van desde grupos del crimen organizado que buscan monetizar las brechas de seguridad, hasta hacktivistas que buscan representar las organizaciones que consideran inmorales o contrarias al interés público. Cualquiera que sea la motivación y quien quiera que sea el atacante, una gran cantidad de ataques son organizados y llevados a cabo por personas calificadas, a menudo con financiamiento.

Este cambio en el panorama ha llevado a muchas organizaciones a participar en un juego de actualización de Seguridad de la Información, ya que a menudo se dan cuenta de que en su programa de seguridad de la información no han recibido el respaldo ejecutivo que se requería o simplemente nunca existió en primer lugar. Estas organizaciones buscan corregir esto y comenzar el camino para iniciar o madurar sus esfuerzos de seguridad de la información, sin embargo hay un problema, la seguridad de la información es una industria que actualmente se encuentra en un período de desempleo negativo; es decir, que hay más posiciones abiertas que candidatos para llenar esas posiciones. Contratar personas es difícil, y contratar personas buenas es más difícil. Para aquellos que buscan empleo, esto puede ser una situación ventajosa; sin embargo, es un alto riesgo para los empleadores que buscan contratar a alguien para una posición de seguridad de la información, ya que estarían brindando una cierta cantidad de confianza con posibles activos de alto valor a un nuevo empleado.

Por esta razón, muchas empresas que ahora están iniciando su programa de seguridad de la información han tomado la ruta para promover a alguien de otro rol, como un administrador del sistema o un arquitecto, a un rol de profesional de la seguridad de la información. Otra práctica común es contratar a un profesional de seguridad de la información más subalterno para un rol del que normalmente sería el caso, y esperar que el empleado recién nombrado aprenda sobre el trabajo.

Crear o mejorar un programa de seguridad puede ser una tarea desalentadora. Con tantas facetas a considerar, cuanto más pensamiento y planificación inicial se ponga en la creación de este programa, más fácil será su administración a largo plazo. Aquí se cubre un esqueleto de un programa de seguridad lógica y los pasos administrativos iniciales.

Para no caer en el hábito de sólo realizar tareas, rutinas o completar la configuración con la mentalidad de "Así es como siempre lo hemos hecho", este tipo de pensamiento solo obstaculizará el progreso y disminuirá la postura de seguridad a medida que pase el tiempo.

**Establecer Equipos**

Equipo Ejecutivo: Una oficina principal de información (CIO) o una oficina principal de seguridad de la información (CISO) proporcionará el apalancamiento y la autoridad necesaria para las decisiones y cambios en toda la empresa. Un equipo ejecutivo también podrá brindar una visión a largo plazo, comunicar los riesgos corporativos, establecer objetivos, proporcionar fondos y sugerir hitos.

Equipo de Riesgos: Muchas organizaciones ya tienen un equipo de evaluación de riesgos, y esto puede ser un subconjunto de ese equipo siendo la seguridad será la prioridad número uno. Este equipo calculará los riesgos que rodean muchas otras áreas del negocio, desde ventas hasta marketing y finanzas. La seguridad puede no ser algo con lo que estén muy familiarizados. En este caso, se les puede enseñar los fundamentos de seguridad caso por caso, o se podría agregar un analista de riesgos de seguridad al equipo. Un marco de riesgo como el marco de evaluación de amenazas, activos y vulnerabilidades operativas y críticas puede ayudar.

Equipo de Seguridad: El equipo de seguridad realizará tareas para evaluar y fortalecer el medio ambiente y está enfocado hacia esto junto al el equipo ejecutivo. Son responsables de las operaciones de seguridad diarias, incluida la administración de activos, la evaluación de amenazas y vulnerabilidades, el monitoreo del entorno para detectar ataques y amenazas, la gestión de riesgos y la capacitación. En un entorno lo suficientemente grande, este equipo se puede dividir en una variedad de subequipos como redes, operaciones, aplicaciones y seguridad ofensiva.

Equipo Auditor: Siempre es una buena idea tener un sistema de controles y balances. Esto no es solo para buscar brechas en los procesos y controles de seguridad, sino también para garantizar que se cubren las tareas y los ambitos correctos.

**Evaluar amenazas y riesgos**

La evaluación de amenazas y riesgos será increíblemente diferente para cada organización. Cada huella interna y externa es única cuando se combina con la infraestructura individual involucrada. La evaluación de estos incluye una descripción general de alto nivel, así como un profundo conocimiento de los activos. Sin el conocimiento de las amenazas y los riesgos a los que se enfrenta su organización, es más difícil adaptar a la medida las tecnologías y las

recomendaciones para proporcionar una defensa adecuada. La gestión de riesgos a menudo se divide en cuatro pasos: identificar, evaluar, mitigar y monitorear

La línea de base de la seguridad de la organización es solo un paso más en esa gestión. Los artículos que se deben reunir incluyen:

• Políticas y procedimientos
• Puntos finales: escritorios y servidores, incluida la fecha de implementación y la verión del software
• Licencias y renovación de software, así como certificados SSL.
• Huellas en Internet: dominios, servidores de correo, dispositivos dmz
• Dispositivos de red: enrutadores, conmutadores, puntos de acceso, IDS / IPS y tráfico de red
• Registro y seguimiento.
• Puntos de ingreso / egreso: contactos ISP, números de cuenta y direcciones IP
• Proveedores externos, con o sin acceso remoto, y contactos principales

**Identificaciones**

No solo se deben identificar las amenazas específicas de la industria, sino también las amenazas de tendencias generales, como malware, ransomware, phishing y ataques remotos. Dos lugares muy importantes para tomar nota son el control de seguridad crítico de OWASP top 10 y CIS 20 (anteriormente conocido como SANS Top 20).

**Evaluaciones**

Una vez que se hayan identificado los riesgos potenciales, evalúe estos riesgos para determinar si se aplican al entorno en particular. Las tareas como las exploraciones de vulnerabilidad internas y externas, las auditorías de reglas de firewall y la administración y descubrimiento de activos brindarán una visión más amplia del tipo de exposición al riesgo general.

**Mitigación**

La mitigación de riesgos es la carne y los huesos de por qué todos estamos aquí; También es el propósito de la mayoría de este libro. Las opciones incluyen evitar, remediar, transferir o aceptar el riesgo.

**Monitoreo**

Realice un seguimiento del riesgo a lo largo del tiempo con reuniones trimestrales o anuales programadas. A lo largo del año, se habrán producido muchos cambios que afectarán la cantidad y el tipo de riesgo que debe considerar. Como parte de cualquier monitoreo de cambio o control de cambio, determine si el cambio está afectando el riesgo de alguna manera.

**Priorizaciones**

Una vez que se han identificado y evaluado las amenazas y los riesgos, también se deben priorizar desde el porcentaje de riesgo más alto al más bajo para la remediación, con una concentración en la protección continua. Sin embargo, esto no siempre tiene que ser una tarea costosa. Se puede realizar una gran cantidad de mitigaciones defensivas con un costo mínimo o nulo para una organización. Esto permite muchas oportunidades para iniciar un programa de seguridad sin tener un presupuesto para hacerlo. La realización de la diligencia debida requerida para que el programa despegue de forma gratuita debería hacer hablar a un equipo ejecutivo.

**Temas y Políticas**

Para facilitar la lectura, la actualización y la administración general, probablemente sea más fácil producir un conjunto de documentos de políticas en lugar de un solo documento monolítico. La selección de cómo se dividen las políticas es, por supuesto, una cuestión de determinar qué es lo más apropiado para su organización. Puede tener un marco de seguridad favorito, como ISO 27002, por ejemplo, del cual puede inspirarse. De manera similar, alinear los temas de políticas con un régimen de cumplimiento regulatorio particular puede estar más alineado con los objetivos de su organización. En realidad, hay muchas similitudes de alto nivel entre muchos de los marcos:

Política de cifrado aceptable
Política de uso aceptable
Política de escritorio limpio
Política del plan de recuperación de desastres
Política de aceptación de firma digital
Política de correo electrónico
Politica etica
Política de planificación de respuesta pandémica

Pautas para la construcción de contraseñas
Política de protección de contraseña
Política de plan de respuesta de seguridad
Política de protección de clave de cifrado de usuario final
Política de evaluación de adquisiciones

Política de requisitos de línea base de Bluetooth
Política de acceso remoto
Política de herramientas de acceso remoto
Política de seguridad de enrutadores y conmutadores
Política de comunicación inalámbrica
Estándar de comunicación inalámbrica
Política de credenciales de base de datos
Política de eliminación de equipos.
Estándar de registro de información
Política de seguridad de laboratorio
Política de seguridad del servidor
Política de instalación de software
Política de seguridad de la estación de trabajo
Política de seguridad de la aplicación web

**Almacenamiento y Comunicaciones**

La naturaleza de las políticas y los procedimientos tiene por objeto prestar la mayor comunicación estándar posible a la organización en su conjunto. Para hacer esto, las políticas deben ser fácilmente accesibles. Hay muchos paquetes de software que no solo pueden proporcionar una interfaz web para políticas, sino que también tienen procesos integrados de revisión, control de revisión y aprobación. El software con estas características lo hace mucho más fácil cuando hay una multitud de personas y departamentos que crean, editan y aprueban políticas.

Otra buena regla general es que, al menos una vez por proceso de revisión, tenga dos copias de todas las políticas impresas. Como la mayoría de ellos se utilizarán en formato digital, habrá muchas políticas que se refieren y están directamente relacionadas con el tiempo de inactividad o los procedimientos de recuperación de desastres. En casos como estos, puede

que no sean accesibles a través de medios digitales, por lo que es mejor tener una copia de seguridad en forma física.

## CONCLUSIONES

El campo de la tecnología de la información ya no es nuevo, y ha llegado el momento de que la educación se centre en producir productos de calidad de forma más rápida y económica y con muchas alternativas nuevas sobre cómo gestionar y modelar un sistema utilizando herramientas de análisis sofisticadas y prácticas de gestión avanzadas, así como enfatizar el cómo y cuándo pueden aplicarse mejor y qué beneficios pueden derivarse de su aplicación.

Hay una serie de temas nuevos que son parte integral del proceso de desarrollo de software, incluidas áreas como ciber seguridad, big data y la transformación digital, así como interactuar directamente con los usuarios, pero el verdadero desafío siempre se aborda en el cómo los analistas necesitan pronosticar o predecir lo que necesitarán los usuarios en el futuro. Por ejemplo, los analistas deberán asumir riesgos y comprender que los requisitos predictivos que puedan desarrollar, tendrían una vida útil relativamente corta, antes de que nuevas necesidades de los usuarios o las leyes mismas los puedan vuelven obsoletos y desafortunadamente no existen soluciones instantáneas para predecir exitosamente si se producirán fallas o no.

**RUBRICA**

## ESTANDARES DE PROGRAMACIÓN

i. Generalidades

| | |
|---|---|
| Proyecto | Sistema transaccional |
| Unidad | Ingeniería de Diseño |
| Tipo de entrega | Grupal |
| Fecha entrega | 24/09/2020 |
| Modo de entrega | En archivo PDF. La portada del archivo debe de indicar los integrantes del grupo de trabajo. |
| Nombre de tarea | Diseño de sistemas |
| Ponderación | 40% |

ii. Descripción de la actividad

En la propuesta de sistema transaccional, se pide realizar el diseño del sistema considerando:

1. Diseño de datos
    a. Diagrama relacional
    b. Diccionario de datos
2. Diseño de interfaces
    a. Patrones de diseño
    b. Diseño de interfaces de entrada, salida y reportes
3. Diseño de procesos
    a. Flujos de procesos
4. Diseño de arquitectura
    a. Diagrama arquitectónico
    b. Descripción de cada componente
5. Diseño de seguridad
    a. Seguridad física
    b. Seguridad lógica

iii. Rubrica

| Criterio | Porcentaje | Nota | Observación |
|---|---|---|---|
| Diseño de datos | 20% | | |
| Diseño de interfaces | 20% | | |
| Diseño de procesos | 20% | | |
| Diseño de arquitectura | 20% | | |
| Diseño de seguridad | 20% | | |
| Total | 100% | | |