

CAPITULO VI

PRUEBAS DEL SOFTWARE

El objetivo de la realización de pruebas es encontrar el mayor número posible de errores con una cantidad de esfuerzo racional a lo largo de un espacio de tiempo realista.

Para probar adecuadamente los sistemas orientados a objetos hacemos dos cosas:

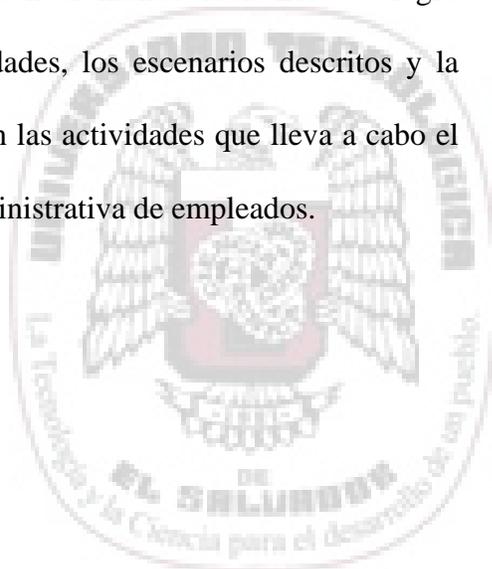
- Corrección del análisis y diseño orientado a objetos.
- El desarrollo de red de clases y colaboraciones.

6.1 Corrección del Análisis y Diseño Orientado a Objetos.

Los modelos de Análisis y Diseño no pueden probarse en sentido convencional pues no pueden ejecutarse. Sin embargo, pueden usarse las revisiones técnicas formales para examinar la corrección (exactitud) y consistencia de ambos modelos.

Verificación de exactitud del modelo

Para verificar la exactitud comparamos el modelo con el mundo real. Esto se logró revisando los casos de uso, las secuencias de actividades, los escenarios descritos y la identificación de las clases para luego compararlos con las actividades que lleva a cabo el Departamento de Recursos Humanos en la gestión administrativa de empleados.



Encontramos algunas inexactitudes en la documentación las cuales fueron corregidas en su momento, así como todas las repercusiones que se habían generado en el modelo completo.

Verificación de la consistencia en el modelo

Un modelo consistente es aquel que tiene representaciones correctamente reflejadas en todas las partes del modelo. Bajo este concepto revisamos el modelo de análisis y diseño utilizando las representaciones CRC (Clase-Responsabilidad-Colaboraciones), que nos ayudó a determinar las relaciones y operaciones que cada clase debería poseer.

Verificación del buen funcionamiento del software

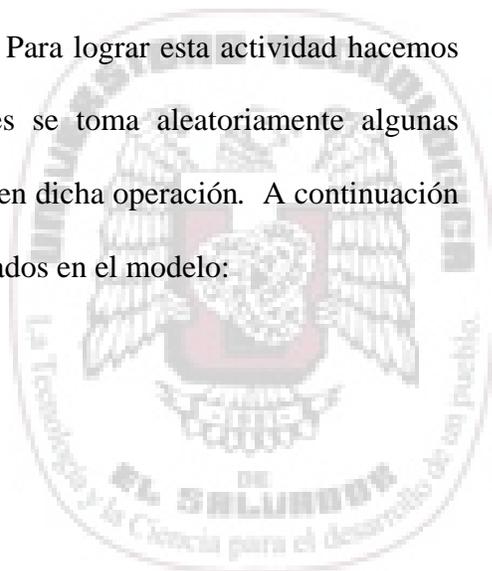
En las pruebas realizadas con el software nos centramos en la depuración de errores lógicos y de sintaxis. Para este propósito utilizamos las diversas cargas de datos con sus posibles fuentes de información y verificamos la validación de cada dato que entra y sale de la aplicación.

6.2 Desarrollo de red de clases y colaboraciones

Cuando comienza la integración del sistema orientado a objetos también se deben comenzar las pruebas de colaboraciones entre clases. Para lograr esta actividad hacemos uso de los *casos de prueba aleatorio* en los cuales se toma aleatoriamente algunas relaciones entre clases y la mensajería que es utilizada en dicha operación. A continuación se presentan algunos casos de prueba que fueron realizados en el modelo:

Caso prueba aleatoria #1.

Clase: RealizarExamen.



Operación: Programar y realizar examen.

Verifica código de aspirante(BancoCurriculo) · Verifica código de plaza de aspirante(PlazaAspirante) · Verifica código de técnico de personal(Empleados).

Caso prueba aleatoria #2.

Clase: Empleados.

Operación: Contratar empleado.

Verifica código de cargo(Cargos) · Verifica código presupuestario(Cifrados) · Toma lugar de trabajo(DeptoysServicios) · Especifica funciones(Funciones) · Toma plaza de ley de salarios(LeySalario).

Caso prueba aleatoria #3.

Clase: Evaluaciones

Operación: Evaluar desempeño del empleado.

Verifica código del empleado(Empleados) · Toma código de factor de evaluación(FactorEvalua) · Toma el valor del factor(ValorEvalua).

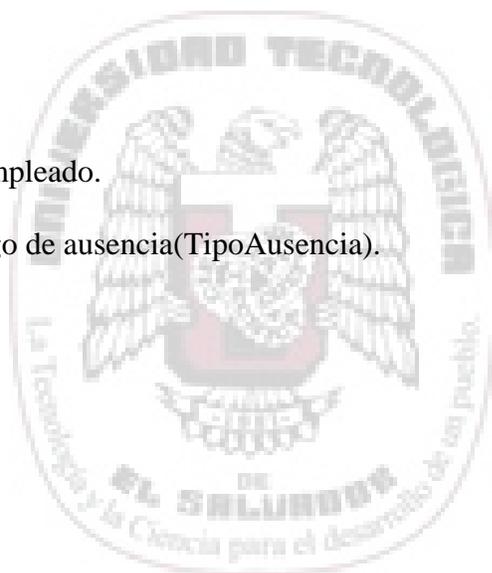
Caso prueba aleatoria #4.

Clase: LicenciaSGoce.

Operación: Otorgar licencia con goce de sueldo a un empleado.

Verifica código de empleado(Empleados) · Toma código de ausencia(TipoAusencia).

Caso prueba aleatoria #5.



Clase: Capacitaciones.

Operación: Capacitar al personal.

Verifica código de empleado(Personal) · Verifica código de escuela(Escuela) · Verifica código del curso(Cursos).

Caso prueba aleatoria #6

Clase: Sanciones

Operación: Establecer sanciones a un empleado.

Toma el código del empleado(Empleados) · Verifica código tipo sanción(TipoSanciones).

Caso prueba aleatoria #7

Clase: RequisiciónPersonal.

Operación: Requerir personal nuevo.

Verifica el código de jefe de departamento(Empleados) · Verifica código de departamento(DeptosyServicios) · Verifica código de disciplina(Disciplinas).

Caso prueba aleatoria #8

Clase: Participante.

Operación: Realizar eventos deportivos o sociales.

Verifica código de empleado participante(Personal) · Verifica código de evento(Eventos).



Los resultados de las pruebas realizadas al análisis, diseño y software del sistema, permitieron ajustar algunas partes del modelo. La documentación presentada en estas fases del desarrollo ya contiene estos cambios.

El siguiente capítulo nos muestra un plan de implementación del sistema presentado por el grupo desarrollador.

