

## **Clase 7 — Contratos informáticos I.**

### **Desarrollo, mantenimiento, implementación, soporte, SLA y responsabilidad por errores de software.**

#### **1. Objetivos de aprendizaje**

Al finalizar la clase, cada estudiante debería poder:

- Diferenciar desarrollo, implementación, mantenimiento y soporte.
- Identificar qué prestación se prometió y cómo se verifica su cumplimiento.
- Distinguir obligaciones de medios y de resultado sin clasificar automáticamente toda la programación en una sola categoría.
- Leer un SLA e identificar métricas, exclusiones, tiempos y remedios.
- Separar bug, incumplimiento y daño indemnizable.
- Reconocer los presupuestos de responsabilidad civil y las principales defensas.
- Evaluar garantías, defectos ocultos y límites de responsabilidad según el contrato aplicable.
- Identificar y preservar evidencia técnica útil para resolver un conflicto.

---

#### **2. Apertura — Un bug no es solamente un problema técnico**

## ¿Qué es un bug?

Un *bug* es un defecto en el diseño, el código, la configuración o la integración de un sistema que hace que se comporte de manera distinta de la esperada en determinadas condiciones. En términos simples: el sistema recibe ciertos datos o instrucciones y produce un resultado incorrecto, incompleto o no previsto.

Por ejemplo, una aplicación de comercio electrónico debería registrar un solo pago aunque, por un problema de conexión, el cliente toque dos veces el botón de compra. Si el sistema procesa ambas solicitudes como operaciones independientes y cobra dos veces, hay un comportamiento defectuoso que debe investigarse.

Un bug no siempre se manifiesta en todo momento. Puede aparecer solo con ciertos datos, una combinación específica de acciones, muchos usuarios conectados al mismo tiempo o una versión determinada del software. Por eso, que una falla no se reproduzca en una prueba posterior no demuestra necesariamente que nunca haya ocurrido. Para analizarla, interesa reconstruir qué versión estaba activa, qué acciones se realizaron, qué datos intervinieron y qué registraron los sistemas.

También conviene distinguir el *bug* de otros conceptos relacionados:

- **Error humano:** una persona realiza una acción incorrecta, por ejemplo, ingresa mal un dato. Esa acción puede revelar o activar un bug, pero no es lo mismo que el defecto del sistema.
- **Incidente:** es el evento que afecta el servicio, como la interrupción de una función o la duplicación de una operación.
- **Falla o defecto:** describe el comportamiento no conforme que se observa; en una conversación cotidiana suele usarse como sinónimo de bug.

- **Cambio de alcance:** es una funcionalidad nueva o modificada que no estaba incluida en lo pactado. Que el sistema no la tenga no significa, por sí solo, que exista un bug.
- **Vulnerabilidad:** es una debilidad que podría permitir un uso indebido o comprometer la seguridad. Puede originarse en un bug, aunque no todo bug constituye una vulnerabilidad.

Para determinar si algo es un bug no alcanza con decir que “el sistema no funciona como esperábamos”. Hay que identificar qué comportamiento se esperaba y de dónde surge esa expectativa: un requisito, una especificación, un criterio de aceptación, una norma técnica o una prestación acordada. Después se compara ese comportamiento esperado con lo que efectivamente hizo el sistema.

En un análisis técnico suelen preguntarse: ¿qué función falló?, ¿en qué versión?, ¿bajo qué condiciones?, ¿se puede reproducir?, ¿qué datos y registros permiten reconstruir lo ocurrido?, ¿qué impacto tuvo y cómo se corrigió? En un análisis jurídico se agrega otra pregunta: ¿ese comportamiento incumplió una obligación asumida? Un bug es, primero, una descripción de un problema técnico. No demuestra por sí solo quién es responsable, si hubo incumplimiento indemnizable ni qué daños deben repararse.

**Idea central:** que exista un bug permite iniciar una investigación; para llegar a una conclusión técnica y jurídica hay que precisar el comportamiento esperado, reconstruir lo ocurrido y vincularlo con las obligaciones aplicables.

## Inicio de la clase

En la clase anterior vimos cómo se forma un contrato y cómo una aceptación digital puede producir obligaciones. Hoy vamos a avanzar: una vez que existe el contrato, ¿qué tiene que hacer cada parte?, ¿cómo sabemos si cumplió?, ¿qué pasa si el sistema falla?

Imaginemos que una empresa contrata una aplicación para cobrar ventas. El viernes a la tarde se despliega una actualización. Durante tres horas no se pueden completar pagos. Algunos clientes abandonan la compra, otros pagan dos veces y el equipo técnico trabaja hasta la noche para restablecer el servicio.

Desde informática, podemos hablar de un incidente, una regresión, un problema de concurrencia o una integración defectuosa. Desde el Derecho, las preguntas son otras, aunque necesitamos la información técnica para responderlas:

- ¿Quién contrató con quién?
- ¿Qué funcionamiento se prometió?
- ¿Quién decidió el despliegue?
- ¿Qué controles debían realizarse?
- ¿Existía un SLA?
- ¿Qué daños ocurrieron realmente?
- ¿Esos daños fueron causados por esta falla?
- ¿Había medidas razonables para evitar o reducir el perjuicio?

La idea central de hoy es esta: un bug no genera automáticamente una indemnización, pero tampoco desaparece jurídicamente porque alguien diga que todo software tiene errores.

### **Pregunta disparadora**

Una persona desarrolladora dice: «Yo cumplí porque entregué el código». El cliente contesta: «No cumpliste porque el sistema no sirve para facturar». ¿Quién tiene razón?

Respuesta inicial: falta conocer lo pactado. Entregar archivos no equivale necesariamente a entregar una solución funcional; pero tampoco puede exigirse

una funcionalidad que nunca integró el alcance. El contrato, sus anexos y la conducta de las partes permiten reconstruir la obligación.

---

### **3. Qué es un contrato informático**

Un contrato informático es un acuerdo que organiza prestaciones relacionadas con bienes o servicios informáticos: desarrollo, implementación, mantenimiento, soporte, licenciamiento, infraestructura o procesamiento de información, entre otras.

No existe una única categoría legal que resuelva todos esos negocios. Según su contenido, pueden intervenir contratos de obra, servicios, licencias y combinaciones de prestaciones. Que una operación se denomine «contrato informático» no significa que todas sus prestaciones sean atípicas.

Los arts. 1251 y 1252 del Código Civil y Comercial de la Nación —CCCN— establecen las bases para distinguir los contratos de obra y de servicios.

#### **Artículo 1251 — Definición**

“Hay contrato de obra o de servicios cuando una persona, según el caso el contratista o el prestador de servicios, actuando independientemente, se obliga a favor de otra, llamada comitente, a realizar una obra material o intelectual o a proveer un servicio mediante una retribución.

El contrato es gratuito si las partes así lo pactan o cuando por las circunstancias del caso puede presumirse la intención de beneficiar.”

#### **Artículo 1252 — Calificación del contrato**

“Si hay duda sobre la calificación del contrato, se entiende que hay contrato de servicios cuando la obligación de hacer consiste en realizar cierta actividad independiente de su eficacia. Se considera que el contrato es de obra cuando se promete un resultado eficaz, reproducible o susceptible de entrega.

Los servicios prestados en relación de dependencia se rigen por las normas del derecho laboral.

Las disposiciones de este Capítulo se integran con las reglas específicas que resulten aplicables a servicios u obras especialmente regulados.”

Fuente: [Código Civil y Comercial de la Nación, Ley 26.994 — texto actualizado oficial, arts. 1251 y 1252.](#)

Para analizar el negocio no alcanza el título del documento. Hay que leer qué se debe hacer, bajo qué condiciones y qué resultado puede exigir el cliente.

#### **4.1. Desarrollo de software a medida**

Se construye una solución conforme a requerimientos de una organización o cliente.

Ejemplo: un sistema para administrar stock de una cadena de farmacias, con integración a facturación, perfiles de acceso y trazabilidad de movimientos.

El alcance debería definir:

- Funcionalidades incluidas y excluidas.
- Requerimientos de rendimiento, seguridad y compatibilidad.
- Entregables: aplicación, código, documentación, pruebas y configuración, según lo acordado.
- Hitos, dependencias y obligaciones del cliente.
- Procedimiento de aceptación.

- Precio y condiciones de pago.
- Gestión de cambios.
- Garantía y corrección de defectos.
- Derechos de uso y titularidad, sin confundir pago con transferencia automática de todos los derechos.

Decir «hacer un sistema de gestión» es insuficiente. ¿Gestionar qué?, ¿con qué volumen?, ¿en qué entorno?, ¿con qué integraciones?

En metodologías ágiles también hay obligaciones jurídicas. Un backlog versionado, criterios de aceptación por historia y acuerdos de alcance de cada sprint pueden concretarlas. Agile no significa ausencia de compromiso ni cambios ilimitados gratuitos.

## **4.2. Implementación**

Consiste en instalar, configurar, parametrizar, integrar o poner en funcionamiento una solución.

Ejemplo: configurar un ERP, migrar datos y capacitar al personal.

Un producto puede funcionar correctamente y, aun así, la implementación ser defectuosa: una migración pierde registros, se asignan permisos excesivos o se configura mal un cálculo.

Hay que definir quién valida los datos de origen, quién realiza la transformación, cómo se concilian los resultados, qué respaldo existe y quién autoriza la salida a producción.

## **4.3. Mantenimiento correctivo**

Busca corregir fallas respecto del funcionamiento comprometido.

Ejemplo: una actualización impide emitir comprobantes que antes podían emitirse correctamente.

No todo pedido del cliente es una corrección. Para distinguir defecto de cambio, se compara el comportamiento observado con la versión y el requisito que debían cumplirse.

#### **4.4. Mantenimiento evolutivo**

Agrega o modifica funcionalidades.

Ejemplo: incorporar un nuevo medio de pago o un módulo de reportes no previsto.

Debería existir un mecanismo de aprobación del alcance, precio, impacto en plazos y criterios de aceptación. Si una adaptación responde a una obligación normativa o funcional ya incluida, no puede calificarse automáticamente como un extra: hay que leer el contrato.

#### **4.5. Mantenimiento preventivo y adaptativo**

El preventivo busca reducir la probabilidad de fallas: revisión de dependencias, monitoreo, pruebas de recuperación o ajustes de capacidad.

El adaptativo adecua el sistema a cambios de entorno: nueva versión del sistema operativo, modificación de una API o cambio de infraestructura.

Conviene establecer cuáles están incluidos. «Mantenimiento» sin definición deja abierto un conflicto sobre qué cubre el abono.

#### **4.6. Soporte**

Es la asistencia para consultas e incidentes. Puede incluir diagnóstico, orientación, escalamiento o intervención técnica, según el acuerdo.

Responder un ticket no equivale a resolverlo. Un contrato debe distinguir recepción, primera respuesta útil, diagnóstico, mitigación, restauración y solución definitiva.

### **Microactividad**

Clasificar estos pedidos y explicar qué documento consultarían:

- «La app dejó de guardar pedidos»: posible corrección.
- «Queremos sumar una billetera digital»: posible evolución.
- «No sé crear un usuario»: soporte o capacitación.
- «La base importada tiene fechas alteradas»: posible falla de implementación.

Las respuestas son orientativas. La clasificación definitiva depende del alcance y de la causa técnica.

---

## **4. La documentación contractual también es una herramienta de ingeniería**

La documentación del proyecto permite aplicar cuatro reglas del Código Civil y Comercial de la Nación —CCCN—: el contrato obliga, debe ejecutarse de buena fe y requiere prestaciones y colaboración de ambas partes. A continuación se distingue el texto legal de su explicación y de los ejemplos informáticos, que son aplicaciones didácticas.

### **Artículo 959 — Efecto vinculante: lo pactado obliga**

“Todo contrato válidamente celebrado es obligatorio para las partes. Su contenido sólo puede ser modificado o extinguido por acuerdo de partes o en los supuestos en que la ley lo prevé.”

Esto significa que el alcance, los entregables, los plazos y el precio acordados no son simples intenciones. Una parte no puede cambiarlos unilateralmente por su conveniencia, salvo que exista una facultad válida o un supuesto legal que lo permita. La regla presupone un contrato válido: no convierte en lícita una cláusula contraria a normas imperativas.

Ejemplo: se acuerda un sistema de stock con tres funcionalidades. El cliente no puede exigir gratuitamente un cuarto módulo solo porque ahora lo necesita. A su vez, el proveedor no puede retirar una funcionalidad comprometida porque resultó más difícil de programar. Primero debe distinguirse si el pedido es un cambio de alcance o la corrección de algo que ya se debía entregar.

En un proyecto ágil, el alcance puede definirse progresivamente mediante un procedimiento acordado. La flexibilidad no equivale a ausencia de obligaciones: hay que identificar quién puede aprobar cambios y qué consecuencias tienen sobre precio, plazo y entregables.

Pregunta para el curso: ¿qué registro permitiría demostrar que una nueva funcionalidad fue efectivamente aprobada y no solamente conversada?

## **Artículo 961 — Buena fe: cumplir no es aprovechar un vacío**

“Los contratos deben celebrarse, interpretarse y ejecutarse de buena fe. Obligan no sólo a lo que está formalmente expresado, sino a todas las consecuencias que puedan considerarse comprendidas en ellos, con los alcances en que razonablemente se habría obligado un contratante cuidadoso y previsor.”

La buena fe exige una conducta leal y coherente durante la negociación, la interpretación y la ejecución. Según el negocio y sus riesgos, permite concretar deberes de información, cooperación y protección aunque no se haya escrito una instrucción para cada situación posible. No habilita, sin embargo, a inventar prestaciones ilimitadas ni a sustituir el alcance por cualquier expectativa subjetiva del cliente.

Ejemplo: el equipo detecta que una integración comprometida dejará de funcionar porque se retirará una versión de la API. Callarlo hasta el día de la entrega no es una conducta leal. Corresponde comunicar el problema, explicar su impacto y analizar las alternativas y responsabilidades según el acuerdo. Esto no determina por sí solo que toda adaptación deba ser gratuita.

También se exige buena fe al cliente: no resulta coherente aprobar los requisitos, impedir las pruebas necesarias y luego atribuir todo el retraso al proveedor sin examinar su propia intervención.

Pregunta para el curso: si el contrato no dice expresamente “avisar cuando aparezca una incompatibilidad”, ¿por qué igualmente puede existir un deber de advertencia?

## **Artículo 1256 — Obligaciones del contratista y del prestador**

Este artículo establece cinco deberes. Se desarrollan por inciso porque cada uno puede traducirse en una práctica concreta del proyecto.

### **Inciso a — Cumplir lo pactado y aplicar conocimientos técnicos razonables**

“ejecutar el contrato conforme a las previsiones contractuales y a los conocimientos razonablemente requeridos al tiempo de su realización por el arte, la ciencia y la técnica correspondientes a la actividad desarrollada;”

No basta entregar algo que tenga el nombre del producto contratado. Hay que respetar los requisitos y aplicar conocimientos técnicos razonablemente exigibles para esa actividad y ese momento. No se exige tecnología perfecta ni una herramienta universal: se evalúan contexto, complejidad, riesgos y compromisos.

Ejemplo: si el módulo debe evitar cobros duplicados ante solicitudes repetidas, su diseño y verificación deben atender ese requisito. Qué pruebas y mecanismos eran exigibles requiere analizar el proyecto; el solo hecho de que exista un bug no demuestra automáticamente negligencia.

Evidencia útil: requisitos versionados, decisiones técnicas, casos de prueba y resultados vinculados a la versión entregada.

#### **Inciso b — Informar sobre el cumplimiento**

“informar al comitente sobre los aspectos esenciales del cumplimiento de la obligación comprometida;”

El cliente necesita información comprensible para decidir. Deben comunicarse los aspectos esenciales: avances relevantes, dificultades, dependencias e impactos que afecten la prestación. Un tablero de tareas puede ayudar, pero no reemplaza una advertencia clara sobre un riesgo decisivo.

Ejemplo: una API no admite el volumen previsto. Informar no es escribir “problema de integración”: es explicar qué función queda afectada, qué alternativas existen y cómo podrían variar el costo o el plazo.

Evidencia útil: reportes, tickets y comunicaciones con destinatario, fecha e impacto identificado.

#### **Inciso c — Proveer materiales adecuados, salvo otra distribución válida**

“proveer los materiales adecuados que son necesarios para la ejecución de la obra o del servicio, excepto que algo distinto se haya pactado o resulte de los usos;”

La norma distribuye la provisión de materiales, pero admite un acuerdo distinto o lo que resulte de los usos. En informática, su aplicación a los recursos del proyecto exige justificar qué se necesita y quién debe aportarlo; no permite afirmar que todo recurso digital sea jurídicamente un material ni que el proveedor deba pagar siempre toda licencia o infraestructura.

Ejemplo: el anexo puede asignar al cliente la contratación del entorno cloud y al proveedor la preparación del entorno de desarrollo. Deben identificarse disponibilidad, costos, permisos y condiciones de uso, sin presumir que están incluidos.

Evidencia útil: inventario de recursos y matriz de responsabilidades.

#### **Inciso d — Usar diligentemente lo aportado y advertir defectos**

“usar diligentemente los materiales provistos por el comitente e informarle inmediatamente en caso de que esos materiales sean impropios o tengan vicios que el contratista o prestador debiese conocer;”

Cuando el cliente aporta recursos comprendidos en esta regla, el proveedor debe utilizarlos diligentemente y advertir de inmediato los problemas que debiese conocer. No se le atribuye conocimiento de cualquier defecto indetectable. Para datos, accesos o entornos digitales, también deben analizarse las obligaciones específicas asumidas y los deberes de información y buena fe, sin equiparlos automáticamente con materiales.

Ejemplo didáctico: el cliente entrega una base para migrar y el equipo detecta identificadores repetidos. Corresponde advertir el riesgo, documentar la

incidencia y acordar cómo validar o corregir los datos, no continuar silenciosamente y atribuir después cualquier pérdida al cliente.

Evidencia útil: validaciones de origen, advertencias y decisiones sobre corrección, respaldos y autorización de la migración.

#### **Inciso e — Ejecutar en el tiempo debido**

“ejecutar la obra o el servicio en el tiempo convenido o, en su defecto, en el que razonablemente corresponda según su índole.”

Si hay un plazo acordado, es la referencia para evaluar el cumplimiento. Si no lo hay, la prestación no queda sin límite temporal: debe determinarse un tiempo razonable según su naturaleza. Una estimación, un hito contractual y un objetivo interno no son necesariamente equivalentes; importa cómo fueron acordados.

Ejemplo: si faltan credenciales del cliente, debe identificarse qué actividad quedó impedida, cuándo se solicitaron y cuánto afecta al cronograma. Esa falta no prueba que todo el proyecto estuviera bloqueado ni autoriza a extender cualquier plazo sin analizar el contrato.

Evidencia útil: cronograma, dependencias, pedidos de acceso y modificaciones aprobadas.

### **Artículo 1257 — Obligaciones del comitente: el cliente también debe cumplir**

#### **Inciso a — Pagar la retribución**

“pagar la retribución;”

En una contratación onerosa, el cliente debe pagar conforme al precio y a las condiciones acordadas. Debe comprobarse si el pago se vincula a un anticipo, un hito, una entrega o una aceptación. La existencia de una controversia no

resuelve por sí sola si puede retenerse un pago: hay que examinar sus fundamentos y el régimen aplicable.

Ejemplo: “pago al entregar” y “pago una vez aprobadas las pruebas de aceptación” describen condiciones distintas. Por eso deben redactarse con precisión.

#### **Inciso b — Proporcionar la colaboración necesaria**

“proporcionar al contratista o al prestador la colaboración necesaria, conforme a las características de la obra o del servicio;”

La colaboración se determina por lo que necesita esa prestación, no por cualquier pedido del proveedor. Puede comprender facilitar accesos, aportar datos, designar responsables y responder validaciones, según el contrato. Debe brindarse por medios seguros: colaborar no significa compartir contraseñas indiscriminadamente ni entregar datos personales innecesarios.

Ejemplo: para integrar una pasarela de pagos puede ser necesario que el cliente habilite un acceso autorizado al entorno de pruebas. Conviene indicar quién lo gestiona, para cuándo y qué actividad depende de él.

#### **Inciso c — Recibir la obra conforme**

“recibir la obra si fue ejecutada conforme a lo dispuesto en el artículo 1256.”

El deber de recepción se refiere a una obra ejecutada conforme a la regla citada. El cliente no puede rechazar arbitrariamente una entrega conforme para exigir funciones nuevas. Pero tampoco debe darse por cumplida la prestación solo porque recibió archivos o pudo iniciar sesión. La conformidad requiere contrastar lo entregado con las obligaciones aplicables.

Recepción y aceptación no son sinónimos: el régimen de verificación y garantía se desarrolla en el apartado 7. Un defecto debe documentarse y valorarse según su importancia, sin convertir cualquier observación menor en una justificación automática para rechazar toda la obra.

### **Síntesis para explicar oralmente**

El art. 959 responde: ¿por qué debo cumplir lo acordado? El art. 961: ¿cómo debo comportarme al cumplirlo? El art. 1256: ¿qué deberes tiene quien ejecuta la obra o el servicio? El art. 1257: ¿qué debe hacer el cliente para pagar, colaborar y recibir una obra conforme?

Aplicación breve: el cliente no entrega el acceso de pruebas y el proveedor detecta el impedimento, pero no lo comunica durante dos semanas. Hay que analizar tanto la colaboración del cliente como la información y la diligencia del proveedor. No alcanza con elegir un culpable: deben reconstruirse las obligaciones, las advertencias y el impacto real sobre las tareas y los plazos.

Fuente de los textos legales: [CCCN, Ley 26.994 – texto actualizado oficial, arts. 959, 961, 1256 y 1257.](#)

En un proyecto, la colaboración del cliente puede incluir entregar accesos, validar requerimientos, aportar datos y responder consultas. Pero la falta de colaboración no exime genéricamente al proveedor: hay que identificar su incidencia y documentar advertencias, pedidos e impactos.

### **Cinco documentos que previenen conflictos**

1. Documento de alcance.
  - Define qué funcionalidades y tareas se incluyen y cuáles quedan fuera del proyecto. También precisa los entregables y evita que cada parte imagine un sistema diferente. Sirve para distinguir una falla respecto de lo prometido de un pedido nuevo.

- Ejemplo: se incluye registrar productos y consultar stock, pero no integrar una tienda online. Si después se solicita esa integración, debe analizarse como un cambio de alcance, no como una corrección.
2. Anexo técnico.
- Describe las condiciones técnicas en las que debe funcionar la solución: entorno, integraciones, versiones compatibles, volumen de usuarios o datos, seguridad y dependencias externas. Permite convertir expresiones generales como “que funcione bien” en condiciones verificables y asignar responsabilidades sobre los recursos necesarios.
  - Ejemplo: se identifica la API de pagos, su versión y el entorno de pruebas, y se establece quién debe proporcionar los accesos para realizar la integración.
3. Plan de entregas.
- Organiza el proyecto en hitos con fechas, entregables y responsables. Debe indicar qué tiene que estar disponible para avanzar y cómo se relaciona cada etapa con el pago, si así se pacta. No basta señalar “primera entrega”: hay que precisar qué se entrega y en qué estado.
  - Ejemplo: primero se presenta un prototipo de pantallas; después, un módulo funcional en pruebas; finalmente, la versión para producción. El prototipo no debe confundirse con el sistema terminado.
4. Protocolo de aceptación.
- Establece cómo se verificará que lo entregado cumple los requisitos: qué pruebas se realizan, con qué datos y en qué entorno, quién las ejecuta y quién valida los resultados. También define cómo se registran y tratan los defectos, distinguiendo los que impiden una función esencial de las observaciones menores.

- Ejemplo: para aceptar el módulo de pagos, se prueba que repetir una solicitud con el mismo identificador no genere un segundo cobro. Se documentan la versión probada, el resultado y las observaciones; recibir los archivos no demuestra por sí solo conformidad.
5. Procedimiento de cambios.
- Define cómo se solicita, evalúa y aprueba una modificación del alcance. Registra su impacto en costo, plazo, integraciones y pruebas, e identifica quién tiene autoridad para aprobarla. Evita que un comentario informal se convierta en una obligación discutida o en trabajo adicional sin condiciones claras.
  - Ejemplo: el cliente pide incorporar una billetera digital. Antes de ejecutarlo, se verifica si ya estaba incluida; si es una prestación nueva, se documentan el trabajo adicional y las condiciones aprobadas. Si se trata de corregir un requisito incumplido, no corresponde tratarlo automáticamente como un extra.

### **Ejemplo de requisito ambiguo y requisito verificable**

Ambiguo: «La aplicación tiene que ser rápida».

Verificable: «En el entorno definido en el anexo, con la carga de prueba acordada, al menos el 95% de las consultas de stock debe responder en menos de dos segundos».

Ese criterio no demuestra por sí solo todo el cumplimiento contractual. Permite verificar una prestación concreta. También debe definirse desde dónde se mide, durante cuánto tiempo y con qué conjunto de datos.

La trazabilidad une requisito, versión, prueba, resultado y aprobación. Una aprobación sin identificar la versión puede no servir para demostrar qué se aceptó.

---

## 5. Obligaciones de medios y de resultado

### 6.1. Obligación de medios

Se debe desarrollar una actividad con la diligencia exigible, sin garantizar un resultado determinado.

Ejemplo: una consultoría exploratoria para identificar posibles mejoras de rendimiento, con metodología, horas y reportes definidos, sin prometer que cualquier arquitectura alcanzará determinada latencia.

No significa «hago lo que puedo». Hay que cumplir la actividad, las técnicas, los deberes de información y la diligencia comprometidos. La entrega del informe o la realización de una prueba pueden, a su vez, ser prestaciones de resultado dentro de esa consultoría.

### 6.2. Obligación de resultado

Se promete un resultado determinado.

Ejemplo: entregar un módulo que calcule una liquidación conforme a las reglas y casos de prueba aprobados.

El art. 1723 CCCN establece que, cuando del acuerdo o de las circunstancias surge que se debe obtener un resultado determinado, la responsabilidad es objetiva. La sola prueba de haber trabajado diligentemente no equivale a cumplimiento del resultado debido.

Esto no elimina el análisis del daño, de la causalidad ni de las defensas legalmente admisibles.

### 6.3. Un proyecto puede reunir ambas

Una misma contratación puede incluir:

- Diagnóstico especializado: actividad diligente, según alcance.
- Entrega de un módulo funcional: resultado definido.
- Primera respuesta a incidentes dentro de un plazo: resultado temporal verificable.
- Investigación de una incompatibilidad externa: actividad y deberes concretos, sin solución garantizada si así fue válidamente delimitada.

La clasificación debe hacerse por prestación. Ni «todo desarrollo es de medios» ni «todo contrato informático garantiza éxito absoluto» son conclusiones correctas.

### 6.4. Diligencia profesional

El art. 1725 CCCN exige valorar especialmente la prudencia y previsibilidad de quien posee conocimientos especiales.

Aplicación didáctica: las exigencias no son iguales para una prueba experimental y para una solución destinada a procesar pagos en producción. Importan el contexto, el riesgo conocido, los compromisos y los recursos disponibles.

El art. 1768 regula profesionales liberales, pero no debe extenderse automáticamente a cualquier persona programadora o software factory. Su aplicabilidad requiere justificar el encuadre; no reemplaza las reglas de resultado, las obligaciones de la empresa ni un eventual régimen de consumo.

---

## 6. Entrega, recepción, aceptación y garantía

No son sinónimos:

- Entrega: se pone a disposición un entregable.
- Recepción: el cliente lo recibe.
- Verificación: se ejecutan las pruebas previstas.
- Aceptación: se reconoce conformidad en los términos acordados y legalmente aplicables.
- Garantía: se cubren defectos o funcionamiento bajo determinadas condiciones.

El art. 1272 CCCN contempla que, si existe un plazo de garantía para verificar la obra o comprobar su funcionamiento, la recepción es provisional y no presume aceptación.

## Ejemplo

Se entrega el módulo de pagos en staging. El cliente verifica veinte casos. Pasan dieciocho y fallan dos que generan cobros duplicados.

La sola entrega no demuestra que el módulo esté conforme. Debe evaluarse la importancia de los defectos y el criterio de aceptación. Tampoco es razonable rechazar sin fundamento una entrega conforme por un pedido nuevo ajeno al alcance.

La aceptación de una versión no implica necesariamente renunciar a defectos ocultos, obligaciones de seguridad o derechos indisponibles. Una cláusula de aceptación por silencio requiere evaluación según el contexto; no debe presentarse como mecanismo universal ni como inmunidad del proveedor.

---

## 7. SLA — Acuerdo de nivel de servicio

Un SLA —Service Level Agreement— define niveles de servicio comprometidos, su medición y las consecuencias de no alcanzarlos.

No es solamente escribir «99,9%». La cifra necesita un método.

### 8.1. SLI, SLO y SLA

- SLI: indicador medido. Ejemplo: porcentaje de pagos válidos completados.
- SLO: objetivo para ese indicador. Ejemplo: lograr determinado porcentaje mensual.
- SLA: acuerdo contractual que define compromisos, medición, condiciones y remedios.

Un objetivo interno no se convierte automáticamente en una garantía contractual. Puede incorporarse si las partes así lo acuerdan.

### 8.2. Disponibilidad

Para un servicio medido por tiempo:

$$\text{Disponibilidad} = [(\text{tiempo computable} - \text{indisponibilidad computable}) / \text{tiempo computable}] \times 100.$$

Ejemplo hipotético: un mes de treinta días tiene 43.200 minutos. Un compromiso de 99,9% admite 43,2 minutos de indisponibilidad, si todo ese tiempo integra el denominador y no hay exclusiones válidas.

Una interrupción computable de 180 minutos arroja aproximadamente 99,5833% de disponibilidad: queda por debajo del objetivo.

Antes de calcular, hay que definir:

- Qué servicio se mide.

- Si se cubre 24/7 o una franja horaria.
- Qué representa una caída y cómo se trata una degradación parcial.
- Qué fuente de monitoreo se utiliza.
- Qué zona horaria y período de medición rigen.
- Qué mantenimiento puede excluirse y con qué aviso.
- Cómo se tratan dependencias externas y controversias de medición.

Un servidor encendido no prueba que los usuarios puedan pagar. Si el proceso crítico falla, el indicador de infraestructura puede ser insuficiente.

### 8.3. Tiempos de respuesta y de restauración

Ejemplo didáctico de severidades:

| Severidad | Situación                                           | Primera respuesta útil | Restauración objetivo |
|-----------|-----------------------------------------------------|------------------------|-----------------------|
| Crítica   | No se pueden completar pagos                        | 15 minutos             | 2 horas               |
| Alta      | Falla un proceso relevante con alternativa limitada | 1 hora                 | 8 horas               |
| Media     | Función secundaria afectada                         | 4 horas hábiles        | 2 días hábiles        |
| Baja      | Consulta o mejora no urgente                        | 1 día hábil            | Según planificación   |

Estos valores no son exigencias legales ni estándares universales: son un ejemplo de negociación. «Restauración objetivo» no debe confundirse con un plazo garantizado si el contrato no le asigna ese carácter.

Hay que determinar cuándo empieza el reloj, qué pausa lo detiene, si se cuentan horas corridas o hábiles y quién puede cambiar la severidad. Un acuse automático no necesariamente constituye respuesta técnica útil.

#### **8.4. Mitigar no es corregir definitivamente**

- Mitigación: reducir el impacto; por ejemplo, desactivar una función defectuosa.
- Restauración: recuperar el servicio; por ejemplo, realizar rollback.
- Corrección definitiva: eliminar la causa raíz y verificar la solución.

Pueden tener plazos distintos. Conviene registrar cada hito.

#### **8.5. Backups, RPO y RTO**

- RPO: punto objetivo de recuperación; expresa la pérdida de datos temporal que se admite como objetivo.
- RTO: tiempo objetivo para recuperar el servicio.

Tener backup no demuestra recuperabilidad. Hay que definir frecuencia, alcance, protección, retención y pruebas de restauración. El contrato debe establecer qué metas se garantizan y en qué escenarios.

#### **8.6. Créditos de servicio, penalidades y daños**

Un crédito de servicio suele consistir en una reducción del abono o del precio cuando no se alcanza un nivel comprometido. En principio, funciona como *remedio contractual*; no equivale automáticamente a una indemnización integral de daños.

Si el contrato prevé una *cláusula penal*, su régimen debe analizarse específicamente: el art. 793 CCCN contempla, para el supuesto que regula, una regla de sustitución de la indemnización por la pena. Por eso no corresponde afirmar —sin leer el texto pactado y su función— que siempre se “acumulan” multa, crédito e indemnización.

Conclusión didáctica: hay que interpretar qué se pactó (crédito, penalidad, tope, remedios exclusivos o acumulables), su alcance y su validez a la luz del régimen aplicable; un crédito pequeño, por sí solo, no vuelve válida una exclusión amplia de responsabilidad si vulnera límites legales.

---

## 8. Cuándo un bug puede generar responsabilidad civil

Para trabajar un caso conviene separar cuatro cuestiones (CCCN, arts. 1716, 1717, 1721, 1724, 1725, 1726, 1727, 1728 y conc.), porque en la práctica suelen confundirse y eso lleva a reclamos mal planteados o defensas incompletas:

1. *Conducta o incumplimiento jurídicamente relevante*: qué obligación concreta se incumplió (o qué deber se vulneró) según el contrato, anexos, buena fe y estándares técnicos razonables.
2. *Factor de atribución*: por qué título se imputa (culpa, dolo, responsabilidad objetiva por resultado, riesgo/vicio, consumo, etc.).
3. *Daño resarcible*: qué perjuicio real, cierto y acreditable se produjo (no meras molestias o conjeturas).
4. *Relación causal*: cómo se conecta ese daño con esa conducta, según causalidad adecuada y previsibilidad contractual.

La clave didáctica es esta: “hubo un bug” describe un hecho técnico; “hay responsabilidad civil” es una conclusión jurídica que exige reconstruir y probar esos cuatro planos.

### 9.1. Conducta o incumplimiento jurídicamente relevante

¿Qué obligación no se cumplió o qué deber se vulneró?

Ejemplos (según lo pactado): entregar un cálculo distinto del requerido, desplegar sin la validación acordada, omitir el monitoreo comprometido, o no atender un incidente dentro del plazo debido (SLA).

La existencia de un error técnico necesita contrastarse con el alcance y el contexto jurídico. Un defecto menor corregible no tiene necesariamente la misma consecuencia que una falla que frustra la función principal del sistema.

### 9.2. Factor de atribución

Es el fundamento por el que el daño se imputa al responsable.

- Culpa: actuación negligente, imprudente o técnicamente imperita respecto de la diligencia exigible. Ejemplo: omitir controles razonables y comprometidos frente a un riesgo conocido.
- Dolo: producción intencional del daño o manifestación de indiferencia por los intereses ajenos en los términos del art. 1724. No toda decisión discutible ni todo bug conocido prueban dolo.
- Factor objetivo por resultado: art. 1723; el análisis no depende solamente de demostrar culpa.
- Riesgo o vicio: arts. 1757 y siguientes, cuando el supuesto pueda encuadrarse en ese régimen.

No todo uso de software es automáticamente una actividad riesgosa ni todo código constituye, sin discusión, una cosa a estos efectos. Debe justificarse el encuadre según la actividad, el daño y las normas aplicables. En una relación de consumo también puede intervenir el art. 40 de la Ley 24.240.

### 9.3. Daño resarcible

No basta la molestia abstracta: hay que individualizar el perjuicio.

Posibles rubros:

- Gastos razonables de recuperación o reparación.
- Costos adicionales de operación.
- Pérdidas económicas efectivas.
- Lucro cesante, si existe base suficiente.
- Pérdida de una chance seria, cuando corresponde.
- Consecuencias no patrimoniales, si se cumplen sus requisitos y legitimación.

No debe presumirse que toda venta no completada representa una ganancia perdida. Facturación bruta, margen, costos evitados y recuperación posterior de ventas son conceptos distintos.

#### **9.4. Relación causal**

¿Este daño fue causado por la falla atribuida?

Ejemplo: si bajaron las ventas, hay que distinguir el efecto de la caída del sistema de otras causas como falta de stock, una promoción cancelada o menor demanda.

Los arts. 1726 y 1728 permiten analizar causalidad y previsibilidad contractual. En contratos, se consideran las consecuencias previstas o previsibles al celebrarlos; el dolo modifica el alcance temporal de esa valoración conforme al art. 1728.

Responsabilidad objetiva no significa responsabilidad sin causalidad ni pago automático de cualquier cifra reclamada.

## **9. Daños y prueba: cómo se construye un reclamo serio**

### **Daño emergente**

Es la pérdida o gasto patrimonial efectivo. Ejemplo: contratación urgente de una recuperación técnica necesaria por el incidente.

Se acredita con comprobantes, pagos, informes y conexión causal. Un gasto ajeno al incidente no se incorpora porque ocurra el mismo día.

### **Lucro cesante**

Es la ganancia frustrada que puede justificarse con suficiente sustento.

Un análisis puede considerar ventas comparables, estacionalidad, margen y operaciones recuperadas después de la restauración. No se multiplica automáticamente cada intento de checkout por el precio total del carrito.

### **Pérdida de chance**

Se afecta una oportunidad seria de obtener un beneficio o evitar un perjuicio. Se indemniza la oportunidad perdida, no como si todo el resultado final hubiera sido seguro.

### **Evitar duplicaciones**

No sumar bajo nombres distintos la misma pérdida. Tampoco reclamar como daño propio dinero de terceros sin justificar qué perjuicio sufrió quien reclama.

Un cobro duplicado puede generar obligación de restitución frente al usuario. Esa restitución, la indemnización de daños adicionales y el eventual reclamo entre comercio y proveedor son cuestiones distintas.

## **Prevención y mitigación**

El art. 1710 CCCN aporta la dimensión preventiva. Ante una falla, no alcanza con discutir quién tiene la culpa: corresponde adoptar medidas razonables para evitar que el daño continúe o se agrave.

Ejemplos: detener nuevos cobros duplicados, activar una alternativa, conservar evidencia y advertir a los responsables pertinentes. La respuesta debe proteger datos personales y no divulgar credenciales o información de usuarios.

---

## **10. Eximentes: un ciberataque no es una respuesta jurídica completa**

### **Caso fortuito o fuerza mayor**

El art. 1730 CCCN define el hecho imprevisible o inevitable y su efecto, con las excepciones aplicables. El art. 1733 contempla situaciones en las que subsiste la responsabilidad.

Que exista un ataque no demuestra por sí solo fuerza mayor. Hay que estudiar su carácter, previsibilidad, medidas exigibles, riesgo asumido e incidencia causal.

Ejemplo: una falla favorecida por credenciales expuestas y controles comprometidos omitidos no se vuelve inevitable por denominarla «hackeo».

### **Hecho del cliente o damnificado**

Puede incidir si el cliente modifica la configuración, entrega datos defectuosos o impide una intervención necesaria.

Debe demostrarse el efecto concreto. No basta decir «el usuario hizo algo mal». También importan las advertencias, permisos, validaciones y medidas razonables de prevención.

### **Hecho de terceros**

El art. 1731 exige, para la exoneración por un tercero por quien no se responde, los caracteres del caso fortuito.

No todo proveedor cloud o subcontratista es automáticamente un tercero ajeno que libera al contratante principal. Deben analizarse las obligaciones asumidas, la cadena de ejecución y el art. 732 sobre actuación de auxiliares.

### **Imposibilidad**

Una dificultad, un costo mayor o una incidencia temporal no equivalen automáticamente a imposibilidad liberatoria. Hay que verificar los requisitos legales, el carácter de la imposibilidad y los riesgos contractualmente asumidos.

### **Idea para repetir en clase**

La causa técnica del incidente y la causa jurídica de exoneración no son lo mismo. Hay que vincularlas y probarlas.

---

## **11. Vicios ocultos y garantías de funcionamiento**

Un defecto oculto no es simplemente cualquier error descubierto después de la entrega. Deben verificarse los presupuestos del régimen: naturaleza del negocio, características del defecto, conocimiento posible y condiciones legales.

El art. 1051 CCCN distingue defectos ocultos y vicios redhibitorios: estos últimos tienen una gravedad que vuelve la cosa impropia para su destino o reduce su utilidad al punto definido por la norma.

En una obra, el art. 1272 remite, para defectos no ostensibles, a los arts. 1054 y concordantes. Esa articulación puede resultar pertinente en un desarrollo; no autoriza a aplicar mecánicamente todas las reglas de transmisión de cosas a cualquier servicio digital.

### **Tres vías que deben distinguirse**

- Incumplimiento contractual: no se realiza lo debido.
- Garantía contractual: cobertura expresamente acordada, dentro de límites legales.
- Saneamiento por defectos ocultos: si se configuran sus presupuestos.

Pueden relacionarse, pero no son intercambiables ni tienen siempre los mismos plazos o remedios.

### **Ejemplo**

Un sistema cumple las pruebas ordinarias, pero ante cierta concurrencia duplica cobros por una condición existente en la versión entregada.

Preguntas: ¿era un requisito cubierto?, ¿el defecto existía al entregar?, ¿podía detectarse mediante un examen adecuado?, ¿qué nivel de concurrencia debía soportar?, ¿hay garantía vigente?, ¿qué régimen corresponde?

El art. 1054 prevé la denuncia del defecto oculto dentro de sesenta días desde su manifestación, con las precisiones y excepciones de la norma. El art. 1055 contiene plazos de caducidad por clases de cosas. No debe enseñarse que todo software tiene automáticamente seis meses de garantía ni confundir caducidad, prescripción y duración de una garantía pactada.

La responsabilidad decenal por determinadas obras de larga duración de los arts. 1273 a 1275 no debe trasladarse automáticamente a un desarrollo informático.

---

## **12. Cláusulas limitativas: hasta dónde puede pactarse**

En una contratación paritaria pueden negociarse distribuciones de riesgos y límites, pero su validez no deriva solamente de una firma.

El art. 1743 CCCN invalida cláusulas que afectan derechos indisponibles, contradicen buena fe o leyes imperativas, son abusivas o liberan anticipadamente del daño doloso del deudor o de quienes debe responder.

También deben revisarse las reglas de adhesión y, si existe consumo, la Ley 24.240. Su art. 37 considera no convenidas las cláusulas que desnaturalizan obligaciones o limitan responsabilidad por daños.

Una persona jurídica puede ser consumidora si reúne el requisito de destinataria final. Tampoco toda compra empresarial de software es de consumo: hay que analizar su finalidad y vinculación con la actividad.

### **Cláusula problemática**

«El proveedor nunca responderá por errores, interrupciones, pérdida de datos o cualquier daño, incluso si se produce por su propia actuación».

La amplitud de la exclusión puede vaciar la prestación y vulnerar límites legales. No se convalida automáticamente porque se use como texto habitual en la industria.

### **Negociaciones que requieren análisis**

- Tope económico relacionado con el precio y riesgo del contrato.
- Exclusión de determinados daños claramente definidos.
- Tratamiento de indisponibilidad de dependencias identificadas.
- Cobertura de incidentes de seguridad y confidencialidad.
- Remedios exclusivos o acumulables.

No corresponde recomendar un tope universal ni importar sin adaptación categorías como «daños indirectos». En derecho argentino hay que precisar qué se pretende excluir y si puede excluirse válidamente.

---

## **13. Cláusulas didácticas para leer y mejorar**

### **A. Alcance y aceptación**

«La entrega comprende las funcionalidades y requisitos del Anexo A, identificado por versión. La aceptación se realizará mediante los casos de prueba del Anexo B. Los defectos críticos que impidan la función principal se documentarán y someterán al procedimiento de corrección acordado».

Preguntas: ¿qué es crítico?, ¿quién prueba?, ¿en qué entorno?, ¿cuánto dura la verificación?, ¿qué ocurre con defectos menores?

### **B. Cambios**

«Toda solicitud que modifique el alcance se registrará con descripción, impacto en costo y plazo y aprobación de representantes autorizados antes de su ejecución».

Pregunta: ¿cómo se distingue un cambio de la corrección de un requisito ya comprometido?

### **C. Incidentes**

«Los incidentes se registrarán por el canal acordado, con severidad, hora y evidencia disponible. La primera respuesta técnica, restauración y corrección definitiva se medirán separadamente».

Preguntas: ¿también inicia el plazo una alerta de monitoreo?, ¿quién escala?, ¿qué pasa fuera del horario de cobertura?

### **D. Dependencias**

«Las dependencias externas, sus responsables y los requisitos de acceso se identificarán en el anexo técnico. Los impactos sobre el servicio se documentarán y comunicarán sin demora indebida».

Pregunta: ¿identificar una dependencia basta para excluir toda responsabilidad? No. Hay que examinar la obligación principal, prevención y validez de la distribución del riesgo.

### **E. Evidencia**

«Ante un incidente se preservarán, dentro de las obligaciones de confidencialidad y protección de datos, versiones, registros y resultados de pruebas pertinentes, documentando origen, fecha, responsables y controles de integridad».

Pregunta: ¿un hash prueba por sí solo quién provocó el incidente? No; ayuda a controlar integridad de una pieza, pero no acredita automáticamente autoría, causalidad ni veracidad originaria.

## 14. Doctrina y jurisprudencia verificadas para la preparación docente

### 15.1. Doctrina

Herrera, Marisa; Caramelo, Gustavo; Picasso, Sebastián —directores—, Código Civil y Comercial de la Nación Comentado, tomo IV, Infojus, 2015.

Se consultaron los comentarios pertinentes en el texto disponible en la Facultad de Derecho de la UBA:

- Art. 1252: distinción entre obra y servicios a partir del fin comprometido; actividad como fin o actividad como medio para obtener un resultado entregable.
- Art. 1723: importancia de lo convenido y de las circunstancias para identificar una obligación de resultado; la diligencia no sustituye el resultado debido.
- Art. 1743: autonomía contractual y límites de la dispensa anticipada de responsabilidad.
- Art. 1768: régimen específico de profesionales liberales y su articulación con obligaciones de hacer.

La obra identifica a Sebastián Picasso y Luis R. J. Sáenz como comentaristas de los arts. 1708 a 1756 y a Luis R. J. Sáenz de los arts. 1757 a 1780. Los directores de una obra colectiva no deben presentarse automáticamente como autores de todos sus comentarios.

Estas explicaciones son doctrina interpretativa, no una ley ni una solución específica para toda contratación de software. La aplicación a ejemplos informáticos es una construcción didáctica.

[Consultar el tomo IV en la UBA.](#)

## 15.2. CSJN — Santa Coloma

Identificación: «Santa Coloma», Fallos: 308:1160. La Secretaría de Jurisprudencia de la CSJN recoge el precedente en su suplemento sobre reparación plena del daño.

Qué aporta: fundamento general de la reparación y rechazo de reducciones carentes de justificación suficiente; permite explicar por qué la reparación no debe volverse ilusoria frente al perjuicio jurídicamente resarcible.

Qué no aporta: no es un caso sobre software ni determina que cualquier bug genere responsabilidad objetiva, que todo tope contractual sea inválido o que exista daño automáticamente.

Uso docente: valor orientador como principio general; para resolver un contrato de software deben analizarse las obligaciones, causalidad, daños y normas del caso.

Alcance de la verificación: se consultó la sistematización oficial de la CSJN, no el texto íntegro del fallo. La copia localizada en un aula universitaria requirió acceso. Por ese motivo no se incluyen citas textuales del fallo ni números de considerando, ni se presenta como lectura integral.

[Suplemento oficial de la CSJN: Principio de la reparación plena del daño.](#)

No se incorporan precedentes supuestamente específicos de fallas de software cuya identificación y fundamentos no pudieron corroborarse. Tampoco se agregan fallos internacionales ajenos al problema contractual solo por prestigio.

---

## 15. Actividad integradora — Incidente en una pasarela de pagos

## Caso hipotético

Comercio Delta contrata a Código Sur para desarrollar e implementar un módulo de checkout integrado a una pasarela externa. El contrato incluye mantenimiento correctivo y soporte crítico 24/7.

El anexo exige que repetir una solicitud con el mismo identificador no genere un cobro nuevo y prevé pruebas de concurrencia. El cliente debe facilitar accesos al entorno de pruebas.

El SLA exige disponibilidad mensual de 99,9%, primera respuesta útil en quince minutos y restauración en dos horas para incidentes críticos. La caída de la pasarela externa solo se excluye si es la causa acreditada de la indisponibilidad. Las ventanas de mantenimiento deben acordarse y notificarse.

El documento incluye un crédito de servicio, pero no define si sustituye otros reclamos. Otra cláusula dice que el proveedor no responde por «ningún daño producido por errores de software».

El viernes a las 18:00 se despliega una nueva versión. A las 18:10 el monitoreo alerta: algunos pedidos no completan el pago y otros generan cobros duplicados. El cliente avisa a las 18:15. La primera intervención útil queda registrada a las 18:50. El servicio se restaura a las 21:10 mediante rollback.

El proveedor alega una modificación de la API externa. El cliente sostiene que el equipo omitió pruebas. Hay datos de que la API cambió, pero todavía no se determinó si ese cambio causó el incidente ni si fue informado previamente. No hay una pericia concluyente.

El comercio presenta gastos de asistencia extraordinaria, reportes de carritos abandonados y reclamos por duplicación de cobros. Algunos clientes compraron al día siguiente. Se desconoce el margen neto de esas ventas.

## Organización de los 25 minutos

- 12 minutos: análisis en grupos.
- 8 minutos: puesta en común.
- 5 minutos: devolución y corrección de confusiones.

## Consigna

Elaborar una conclusión de una página, distinguiendo hechos, hipótesis y datos faltantes:

1. Identificar las prestaciones y clasificarlas como medios o resultado, justificando.
2. Determinar qué incumplimientos aparecen documentados y cuáles requieren investigación.
3. Evaluar disponibilidad, primera respuesta y restauración.
4. Separar gastos comprobables, posibles ganancias frustradas y restituciones a usuarios.
5. Analizar la defensa basada en la API externa.
6. Examinar la cláusula de exclusión y la función del crédito de servicio.
7. Seleccionar evidencia técnica que debería preservarse.
8. Proponer tres medidas preventivas.

## Guía de resolución

### A. Obligaciones

La entrega de un checkout con el comportamiento de no duplicación acordado presenta una prestación de resultado definida. Las intervenciones de soporte incluyen compromisos temporales verificables. La calificación del proyecto debe hacerse por prestación, no solo por el título del contrato.

### B. Hechos e hipótesis

El comportamiento contrario al requisito y las marcas temporales son indicios relevantes. La omisión de pruebas y la causa exacta necesitan verificación. La existencia de una modificación externa no demuestra por sí sola una eximente.

#### **C. SLA (*Service Level Agreement* (en español, *Acuerdo de Nivel de Servicio*))**

Si la indisponibilidad de 18:10 a 21:10 fue total y computable, son 180 minutos. En un mes hipotético de treinta días, sin otras caídas ni exclusiones, arroja aproximadamente 99,5833%: incumple 99,9%.

Si la afectación fue parcial, no debe tratarse automáticamente como caída total: se aplica el método pactado y se verifica si contempla fallas del proceso crítico.

La primera respuesta útil excede quince minutos tanto desde la alerta —cuarenta minutos— como desde el aviso —treinta y cinco—. La restauración excede dos horas desde cualquiera de esos dos puntos. Falta confirmar cuál activa contractualmente el reloj; en este supuesto no cambia la conclusión sobre exceder los plazos.

#### **D. Daños**

Los gastos extraordinarios necesitan comprobantes y vinculación causal. Los carritos abandonados no equivalen a ganancia íntegra perdida: hay que estudiar conversión probable, margen y compras recuperadas.

Los cobros duplicados exigen regularización y análisis de reclamos individuales. No se suman automáticamente como daño patrimonial propio del comercio sin verificar cómo impactaron en su patrimonio.

#### **E. Defensa externa**

Hay que revisar versión de API, aviso del cambio, compatibilidad, reparto de responsabilidades y controles asumidos. La exclusión pactada exige causa acreditada: no basta nombrar al tercero.

## **F. Exclusión y crédito**

La exoneración de todo daño por errores puede vaciar obligaciones y debe confrontarse con el art. 1743, las reglas de adhesión y el eventual régimen de consumo. No se presume válida.

El crédito requiere interpretación: no puede afirmarse su acumulación o sustitución de daños sin analizar su función y las reglas pertinentes, incluido el art. 793 si opera como cláusula penal.

## **G. Evidencia**

- Contrato, anexos y cambios aprobados.
- Versión desplegada, commits y configuración.
- Registros de CI/CD, pruebas y autorizaciones.
- Métricas, logs y tickets con zona horaria y sincronización verificadas.
- Solicitudes y respuestas de API, identificadores de transacciones y correlación.
- Avisos del proveedor externo.
- Registros del rollback y validación posterior.
- Comprobantes de gastos y datos necesarios para analizar ventas y reintegros.

Preservar originales, documentar extracción y accesos, controlar integridad y minimizar exposición de datos. Una captura aislada puede servir como indicio, pero no reemplaza la reconstrucción técnica del incidente.

## **H. Prevención**

Pruebas de idempotencia y concurrencia; despliegues graduales con rollback; monitoreo de transacciones completas; alertas sobre cambios de API; protocolo de respuesta y delimitación de responsabilidades.

No hay base suficiente para fijar una indemnización definitiva ni imputar dolo. Sí hay elementos para investigar incumplimientos, preservar prueba y actuar sobre el incidente.

---

## 16. Preguntas frecuentes

### **«Si el cliente aprobó, ¿ya no responde el proveedor?»**

No necesariamente. Hay que analizar qué se aprobó, la versión, las reservas, defectos ocultos y obligaciones que permanecen.

### **«Si la plataforma externa falló, ¿queda libre la software factory?»**

No automáticamente. Importan obligación asumida, causalidad, riesgo distribuido y carácter del tercero.

### **«Si no hay contrato escrito, ¿no hay reclamo?»**

No. Puede existir contrato y obligaciones acreditables por otros medios. La ausencia de documentación aumenta la dificultad para reconstruir alcance, aceptación y responsabilidad.

### **«Si corregí el bug, ¿desaparecen los daños?»**

La corrección puede cumplir un deber y evitar agravamiento, pero no elimina necesariamente perjuicios ya causados.

### **«Si todo software tiene errores, ¿puedo excluir cualquier responsabilidad?»**

No. La complejidad del software no habilita a vaciar obligaciones ni desconocer límites legales.

### «¿Se puede resolver el contrato por cualquier falla?»

No automáticamente. Deben examinarse gravedad, remedios disponibles y régimen de resolución; el art. 1084 CCCN requiere incumplimiento esencial en los términos que establece.

---

## 17. Cierre

Lo más importante no es aprender una lista de contratos. Es aprender a transformar una promesa tecnológica en una obligación que pueda entenderse, verificarse y probarse.

Cuando desarrollamos o mantenemos un sistema debemos saber qué prometimos, cómo se mide, qué depende del cliente, qué depende de terceros y qué vamos a hacer si falla.

Un bug es un hecho técnico. El incumplimiento es una conclusión jurídica. La indemnización requiere además daño, atribución y causalidad.

La seguridad jurídica no empieza cuando llega una demanda. Empieza al definir requerimientos, probar, documentar, advertir riesgos y acordar responsabilidades.

### Ticket de salida

Responder en tres frases:

- Una diferencia entre mantenimiento correctivo y evolutivo.
- Por qué tiempo de respuesta no equivale a tiempo de resolución.
- Qué falta demostrar para pasar de «hubo un bug» a «corresponde indemnizar».

### Tarea breve para la próxima clase

Elegir una funcionalidad de un proyecto real o ficticio y redactar:

- Un requisito verificable.
- Un criterio de aceptación.
- Un nivel de servicio con método de medición.
- Una dependencia externa.
- Una medida de prevención y una evidencia de cumplimiento.

No requiere datos reales de usuarios ni credenciales. La fecha de entrega se fija en clase según el cronograma docente.

---

## 18. Síntesis

1. El contrato informático se analiza por prestaciones y no solo por su nombre.
2. Desarrollo, implementación, mantenimiento y soporte no son equivalentes.
3. El alcance y los criterios de aceptación permiten verificar cumplimiento.
4. Medios exige diligencia; resultado exige el resultado comprometido.
5. Un SLA necesita métricas, método, cobertura, exclusiones y remedios.
6. Responder, mitigar, restaurar y corregir definitivamente son hitos distintos.
7. Un bug no demuestra automáticamente culpa, daño ni causalidad.
8. Las obligaciones de resultado pueden generar responsabilidad objetiva sin eliminar los restantes requisitos.
9. Ataques y terceros no son eximentes automáticas.
10. Garantía contractual, vicios ocultos e incumplimiento tienen presupuestos distintos.

11. Las cláusulas de exclusión o límite requieren control de validez.
12. La evidencia técnica y la prevención son parte de la práctica profesional.

## **19. Fuentes normativas y bibliografía**

### **Normativa argentina**

- CCCN: arts. 732 y 774; 790 y 793; 959 y 961; 1033 a 1058; 1084; 1251, 1252, 1256, 1257 y 1272; 1710 y 1716; 1721 a 1728; 1730 a 1733; 1737 a 1740 y 1743; 1757 y 1768. Los artículos se trabajan según el supuesto, no como un bloque aplicable automáticamente.
- Ley 24.240: arts. 1, 4, 37 y 40, cuando exista una relación de consumo.

[CCCN — texto actualizado oficial.](#)

[Ley 24.240 — texto actualizado oficial.](#)

### **Bibliografía y jurisprudencia**

- Herrera, Caramelo y Picasso —directores—, Código Civil y Comercial de la Nación Comentado, tomo IV, Infojus, 2015. Comentarios consultados indicados en el apartado 15.
- CSJN, Secretaría de Jurisprudencia, Principio de la reparación plena del daño, suplemento oficial; referencia a «Santa Coloma», Fallos: 308:1160. Alcance documental de la consulta explicitado en el apartado 15.





