

# DESARROLLO FORMAL DE SOFTWARE

## PRIMERA ENTREGA DEL PROYECTO

### *Análisis y Especificación Formal de Funcionalidades*

|                             |                                                                                                                          |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Unidades</b>             | 1. Fundamentos del desarrollo formal y calidad del software<br>2. Especificación formal y desarrollo basado en contratos |
| <b>Capítulos integrados</b> | 1. Introducción al Desarrollo Formal<br>2. Calidad y construcción<br>3. Fundamentos de especificación formal             |
| <b>Entrega</b>              | Primera entrega del proyecto                                                                                             |
| <b>Fecha máxima</b>         | 1 de octubre de 2026                                                                                                     |
| <b>Modalidad</b>            | Trabajo en grupo máximo 3 personas                                                                                       |
| <b>Valor</b>                | 100% de la calificación de esta entrega                                                                                  |
| <b>Software</b>             | No se requiere software funcional ni código terminado                                                                    |

#### 1. Propósito de la primera entrega

Esta primera entrega integra los aprendizajes fundamentales de los Capítulos 1, 2 y 3. El grupo tomará tres funcionalidades previamente definidas de su proyecto y realizará un primer análisis de correctitud y calidad, para posteriormente expresar sus condiciones y propiedades mediante herramientas de especificación formal.

La entrega es un avance del proyecto, no el producto terminado. El propósito es dejar una base clara que posteriormente permita construir, establecer contratos, verificar, probar y mejorar el software.

#### 2. ¿Qué integra cada capítulo?

| Capítulo                                         | Pregunta orientadora                                                   | Evidencia en la entrega                                                                                                                  |
|--------------------------------------------------|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Capítulo 1. Introducción al Desarrollo Formal    | ¿Qué debe hacer correctamente la funcionalidad y qué podría salir mal? | Descripción de correctitud, posible error/defecto/fallo y casos normal, límite e inválido.                                               |
| Capítulo 2. Calidad y construcción               | ¿Qué características de calidad debemos considerar?                    | Una característica de calidad relevante y una breve consideración sobre confiabilidad, robustez o mantenibilidad.                        |
| Capítulo 3. Fundamentos de especificación formal | ¿Cómo expresamos esas condiciones con precisión?                       | Proposiciones, $\neg$ , $\wedge$ , $\vee$ , predicados, variables, cuantificadores cuando correspondan, formalización y tabla de verdad. |

### 3. Fecha de entrega y socialización

La fecha máxima de entrega es el **1 de octubre de 2026**. Deben estar cargado en la plataforma antes de iniciar la clase.

Como el proyecto integrado todavía no ha sido socializado formalmente, esta entrega se considera un avance. Si posteriormente se realizan ajustes en las funcionalidades, estos podrán incorporarse en las siguientes entregas.

### 4. Ruta del proyecto

**ANALIZAR → ESPECIFICAR → CONSTRUIR → VERIFICAR → PROBAR → MEJORAR**

La primera entrega se concentra en analizar y especificar. La construcción, los contratos, la verificación y las pruebas se desarrollarán progresivamente en los siguientes cortes.

### 5. ¿Qué se debe entregar?

- 1. Identificación del proyecto:** Nombre del proyecto, integrantes, programa y grupo.
- 2. Descripción general:** Breve explicación del sistema y del problema o necesidad que aborda.
- 3. Tres funcionalidades seleccionadas:** Las tres funcionalidades que serán trabajadas en Desarrollo Formal de Software.
- 4. Descripción de cada funcionalidad:** Propósito, usuario o actor, entradas principales y resultado esperado.
- 5. Correctitud y posibles problemas:** Qué debe hacer correctamente, un posible error/defecto/fallo y casos normal, límite e inválido.
- 6. Consideraciones de calidad:** Una característica de calidad relevante y una breve explicación de cómo podría verse afectada.
- 7. Condiciones y propiedades:** Condiciones que deben cumplirse y propiedades que caracterizan el comportamiento esperado.
- 8. Proposiciones y conectores:** Representar condiciones mediante proposiciones y utilizar  $\neg$ ,  $\wedge$  y  $\vee$ .
- 9. Predicados y variables:** Utilizarlos cuando la propiedad dependa de un elemento o dato variable.
- 10. Cuantificadores:** Utilizar  $\forall$  y/o  $\exists$  cuando la condición o propiedad lo requiera.
- 11. Formalización:** Mostrar el paso de una condición en lenguaje natural a una expresión formal e interpretar su significado.
- 12. Tabla de verdad:** Construir una tabla para una expresión proposicional con dos proposiciones.

### 6. Guía breve para desarrollar cada funcionalidad

#### Paso 1. Seleccionar

Elijan tres funcionalidades concretas que ya estén definidas en el proyecto.

#### Paso 2. Analizar correctitud

Pregúntense qué significa que la funcionalidad funcione correctamente y qué errores, defectos o fallos podrían presentarse.

#### Paso 3. Considerar calidad

Identifiquen una característica de calidad especialmente importante para esa funcionalidad y expliquen por qué.

#### Paso 4. Identificar condiciones

Pregúntense qué debe cumplirse para que la funcionalidad pueda realizarse correctamente.

#### Paso 5. Formalizar

Representen las condiciones mediante proposiciones y, cuando corresponda, predicados, variables y cuantificadores.

#### Paso 6. Interpretar

Expliquen en lenguaje natural qué significa cada expresión formal.

#### Paso 7. Tabla de verdad

Construyan una tabla de verdad para una expresión con dos proposiciones.

### 7. Ejemplo orientador – Reservar cupo

Funcionalidad: Reservar cupo.

P = Hay cupos disponibles.

Q = La solicitud es válida.

R = El usuario está registrado.

Correctitud y casos:

- Debe permitir la reserva cuando se cumplen las condiciones establecidas.
- Caso normal: existen cupos y la solicitud es válida.
- Caso límite: queda un solo cupo disponible.
- Caso inválido: no hay cupos o la solicitud no cumple las condiciones.

Calidad:

La confiabilidad y la robustez son relevantes porque la funcionalidad debe responder correctamente ante situaciones esperadas y condiciones límite o inválidas.

Expresiones:

$P \wedge Q$  = Hay cupos disponibles y la solicitud es válida.

$\neg Q$  = La solicitud no es válida.

$P \vee R$  = Hay cupos disponibles o el usuario está registrado.

Predicado:

$\text{UsuarioRegistrado}(x)$  = “x está registrado en el sistema”.

Ejemplo de cuantificador, cuando corresponda:

$\forall x (\text{UsuarioRegistrado}(x) \rightarrow \text{TieneIdentificacion}(x))$

Interpretación: todo usuario registrado tiene identificación.

Tabla de verdad:  $P \wedge Q$

| P | Q | $P \wedge Q$ |
|---|---|--------------|
| V | V | V            |
| V | F | F            |
| F | V | F            |
| F | F | F            |

Interpretación:  $P \wedge Q$  es verdadera únicamente cuando hay cupos y la solicitud es válida.

## 8. Rúbrica de evaluación

| Criterio                                | Descripción                                                                                             | %   | Puntos |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------|-----|--------|
| Identificación y descripción            | Presenta el proyecto, propósito y tres funcionalidades.                                                 | 10% | 10     |
| Correctitud, errores y casos            | Analiza qué debe hacer correctamente e identifica posibles problemas y casos normal, límite e inválido. | 15% | 15     |
| Calidad                                 | Identifica una característica de calidad relevante y la relaciona con la funcionalidad.                 | 10% | 10     |
| Condiciones y propiedades               | Identifica condiciones relevantes y coherentes con el comportamiento esperado.                          | 15% | 15     |
| Proposiciones y conectores              | Representa condiciones y utiliza correctamente $\neg$ , $\wedge$ y $\vee$ .                             | 15% | 15     |
| Predicados, variables y cuantificadores | Los utiliza correctamente cuando la naturaleza de la propiedad lo requiere.                             | 15% | 15     |
| Formalización e interpretación          | Transforma condiciones del lenguaje natural a expresiones formales y explica su significado.            | 10% | 10     |
| Tabla de verdad                         | Construye e interpreta correctamente una tabla de dos proposiciones.                                    | 5%  | 5      |
| Organización y presentación             | Documento claro, coherente, ordenado y bien redactado.                                                  | 5%  | 5      |

## 9. Criterios importantes

- No se calificará la cantidad de código porque esta entrega no exige programación.
- Las expresiones deben corresponder realmente a las condiciones de la funcionalidad.
- No basta con escribir símbolos: cada expresión debe interpretarse en lenguaje natural.
- No es obligatorio utilizar cuantificadores en todas las funcionalidades; deben emplearse cuando corresponda.
- Debe existir coherencia entre funcionalidad, condiciones, análisis de calidad y representación formal.

## 10. Entrega

Fecha máxima: 1 de octubre de 2026 hasta las 9:00am

Se recomienda cargar un único archivo PDF en el AVA institucional, con los nombres de todos los integrantes.

La entrega puede realizarse antes de la fecha máxima.