

# Programación Backend

## Evaluación Sumativa 1

### Instrucciones para la evaluación.

#### Instrucciones

Los estudiantes deberán desarrollar, de forma **individual**, un archivo de programación del lado del servidor utilizando el framework Django. El propósito es aplicar estructuras básicas del lenguaje y paquetes externos para construir una aplicación funcional.

#### Objetivos de Aprendizaje

1. Reconocer e identificar variables y operaciones del lenguaje según requerimientos.
2. Codificar instrucciones con estructuras de decisión y operadores.
3. Utilizar paquetes externos en la codificación.
4. Implementar una aplicación sencilla en Django.

#### Situación de Evaluación

Los estudiantes deben desarrollar un **proyecto en Django** disponible a través de GitHub para su visualización pública. Se espera que los estudiantes sean capaces de dominar los conceptos de Modelo-Vista-Plantilla, arquitectura alternativa al MVC que se aplica desde Django. Esto incluye que los estudiantes sepan cómo implementar: Modelo, Vistas, Urls y Templates desde el framework.

#### Criterios de Evaluación

1. Identifica variables y operaciones del lenguaje, según requerimiento (1.1.1).
2. Codifica instrucciones y estructuras de decisión y operadores, según requerimientos (1.1.2).
3. Codifica instrucciones utilizando paquetes externos, según requerimientos (1.1.3).
4. Implementa una aplicación sencilla en Django, según requerimientos (1.1.4).

**Casos por resolver. Elige uno para realizar en la evaluación.**

**Caso 1: Sistema de Registro de Visitas para una Empresa**

**Contexto:** Una empresa necesita llevar un control digital de las visitas que recibe diariamente.

**Requerimientos:**

- Registrar nombre, RUT, motivo de visita y hora de entrada/salida.
- Mostrar listado de visitas del día.
- Utilizar estructuras de decisión para validar datos.

**Caso 2: Aplicación de Control de Venta Básico**

**Contexto:** Una pequeña tienda desea digitalizar la venta de productos.

**Requerimientos:**

- Registrar el producto a vender con nombre, código, cantidad y precio.
- Permitir actualizar stock y eliminar productos.
- Mostrar listado de productos disponibles.
- Registrar ventas designadas a un rut de cliente
- Guardar datos de cliente si quiere ser cliente habitual
- En caso de que no, solo pedir rut para la boleta

**Caso 3: Agenda de Contactos Personal**

**Contexto:** Un usuario desea una aplicación para gestionar sus contactos personales.

**Requerimientos:**

- Agregar contactos con nombre, teléfono, correo y dirección.
- Buscar contactos por nombre o correo.
- Validar formato de correo electrónico.

Indicadores	Siempre	A veces	Nunca
1. Identifica correctamente variables y operaciones del lenguaje.	6 puntos	4 puntos	1 puntos
2. Utiliza estructuras de decisión y operadores de forma adecuada.	6 puntos	4 puntos	1 puntos
3. Integra paquetes externos en la solución.	6 puntos	4 puntos	1 puntos
4. Implementa una aplicación funcional en Django según requerimientos.	6 puntos	4 puntos	1 puntos
5. Estructura el código de forma clara y ordenada.	6 puntos	4 puntos	1 puntos
6. Comenta el código para facilitar su comprensión.	6 puntos	4 puntos	1 puntos
7. Valida correctamente los datos de entrada.	6 puntos	4 puntos	1 puntos
8. Utiliza correctamente el entorno de desarrollo y herramientas asociadas.	6 puntos	4 puntos	1 puntos
9. Cumple con los requerimientos funcionales del caso seleccionado.	6 puntos	4 puntos	1 puntos

10. Presenta el archivo sin errores de ejecución.	6 puntos	4 puntos	1 puntos
---	----------	----------	----------