

Ampliación de Programación

- **Programación funcional (*Haskell*)**
 - Paradigma complementario a la programación imperativa (*C*, *Java*)
 - Lenguaje semejante al de **Matemáticas**
 - Funciones (**sin efectos laterales**). **Declarativo** (no hay secuencias únicas de instrucciones). Datos **inmutables** y **polimórficos**. Operaciones de **orden superior** y altamente **paralelizables**. Expresiones **lambda**.
 - **Conciso, elegante, robusto**
 - Estructuras infinitas, evaluación perezosa, razonamiento y verificabilidad de programas
- **Desarrollo en *.NET* (*C#*)**
 - Ampliación de POO
 - *C#* mejora muchos aspectos de *C++* y *Java*
 - Interfaces gráficas de usuario
 - Programación funcional (**delegados**, *LINQ*, *TPL*...)
- **Patrones de diseño funcionales**
 - Concurrencia. Funtores, Mónadas, ...



Ampliación de Programación

- **Interés académico**
 1. **Fortalecimiento** de conocimientos de programación, desarrollando programas más expresivos, concisos, robustos y paralelizables
 - *Java 1.8* (2014) incluye varios elementos de programación funcional
 2. Estudio de un entorno **profesional** de desarrollo (*.NET./C#*)
- **Entornos**
 - *Haskell* (IDE → *Visual Studio Code*)
 - **Open source**, consolidado académicamente (más de 20 años), con creciente expansión en otros ámbitos
 - *.NET* (IDE → *Visual Studio*)
 - **Propietario**, Microsoft, ampliamente usado en entornos profesionales
 - Lenguaje *C#*, *Windows Forms*, programación funcional en *C#* y *F#*
- **Metodología docente**
 - Teoría (aula), 2 horas semanales, 1 hora para ejercicio semanal
 - Prácticas (laboratorio, *Haskell* y *C#*), 2 horas semanales
- **Evaluación**
 - Teoría (70%)
 - Evaluación continua → 3 parciales (25%, 35%, 40%)
 - Evaluación no continua → examen final
 - Prácticas (30%)
- **Profesor coordinador**
 - Francisco Saiz