

- Las computadoras paralelas pueden ser construidas a partir de materias primas con componentes baratos.
- Resolver problemas mayores y más complejos:
 - Muchos de los problemas son tan grandes y/o complejos que no es práctico o posible resolverlos con un solo equipo, especialmente teniendo en cuenta que la memoria de la computadora es limitada.
 - Ejemplo: Motores de búsqueda Web o Bases de Datos procesando millones de transacciones por segundo.
- Proveer CONCURRENCIA:
 - Un solo recurso de cómputo sólo puede hacer una cosa a la vez. Recursos informáticos múltiples pueden hacer muchas cosas al mismo tiempo.
- Aprovechar los recursos no locales:
 - El uso de los recursos informáticos en una red de área amplia, o incluso de Internet, cuando los recursos informáticos locales son escasos o insuficientes.
- Hacer mejor uso del HARDWARE PARALELO Subyacente:
 - Las computadoras modernas, incluso las portátiles, son paralelas en su arquitectura con múltiples procesadores/núcleos.
 - El software Paralelo está destinado específicamente para el hardware paralelo con múltiples núcleos, hilos, etc.
 - En la mayoría de los casos, los programas seriales que se ejecutan en las computadoras modernas, "desperdician" el potencial de cómputo.

El futuro:

- Durante los últimos 20 años, las tendencias marcadas por redes cada vez más rápidas, sistemas distribuidos y arquitecturas de computadora con múltiples procesadores (incluso a nivel de escritorio), muestran claramente que el paralelismo es el futuro de la informática.

Información general

¿Quién utiliza la computación en paralela?

Ciencia e Ingeniería:

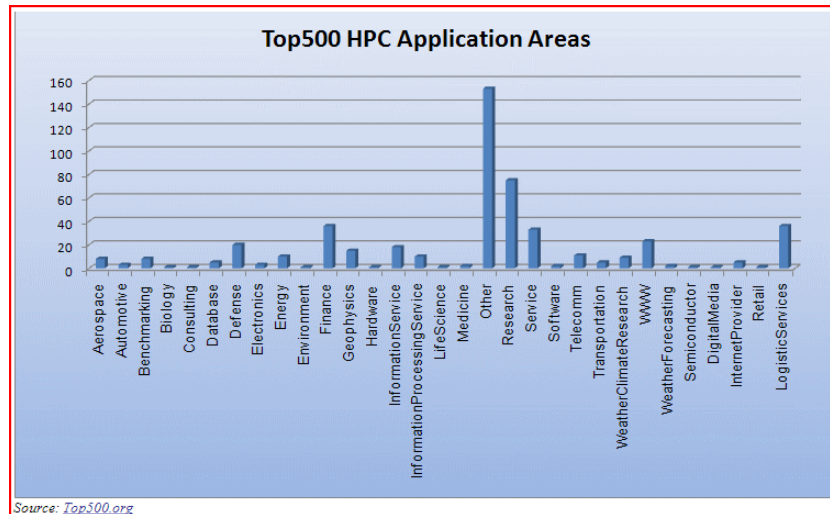
- Históricamente, la computación paralela se ha considerado como "la parte más alta de la informática", y ha sido utilizada para modelar problemas difíciles en muchas áreas de la ciencia y la ingeniería.

Industrial y Comercial:

- Hoy en día, las aplicaciones comerciales proporcionan una fuerza igual o mayor de conducción para el desarrollo de computadoras más rápidas. Estas aplicaciones requieren el procesamiento de grandes cantidades de datos en formas sofisticadas.

Aplicaciones globales:

- La computación paralela ahora se está utilizando ampliamente en todo el mundo, en una amplia variedad de aplicaciones. Por ejemplo:



Conceptos y Terminología

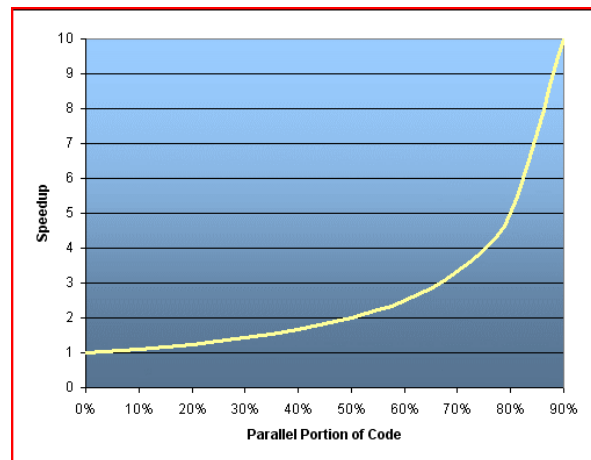
Límites y Costos de Programación Paralela

Ley de Amdahl:

- La Ley de Amdahl afirma que el potencial de aceleración (speedup) de un programa se define por la fracción de código (P) que puede ser paralelizada:

$$\text{speedup} = \frac{1}{1 - P}$$

- Si ninguno de los códigos puede ser paralelizado, $P = 0$ y el $\text{speedup} = 1$ (sin aceleración).
- Si todo el código se paraleliza, $P = 1$ y el aumento de velocidad es infinita (en teoría).
- Si el 50% del código puede ser paralelizado, la máxima aceleración = 2, es decir, el código se ejecutará dos veces más rápido.



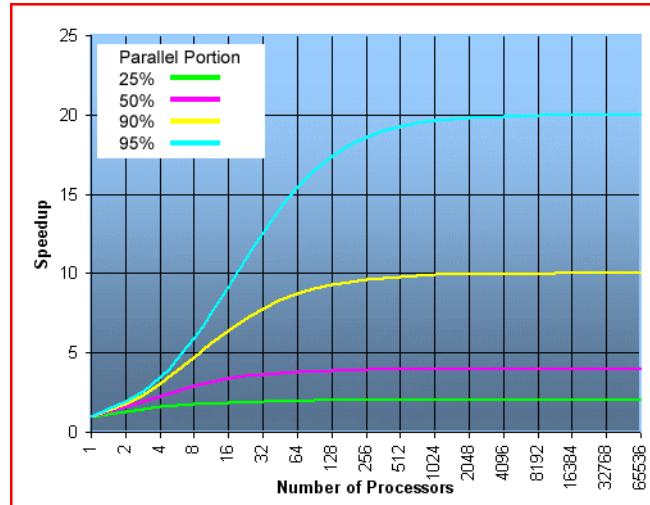
- Al incrementar el número de procesadores que realizan una fracción paralela de trabajo, la relación puede ser modelado por:

$$\text{speedup} = \frac{1}{\frac{P + S}{N}}$$

Donde P = fracción de código paralela, N = número de procesadores y S = fracción de código serial.

- Se hace evidente que hay límites a la escalabilidad de paralelismo. Por ejemplo:

N	speedup		
	P = .50	P = .90	P = .99
10	1.82	5.26	9.17
100	1.98	9.17	50.25
1,000	1.99	9.91	90.99
10,000	1.99	9.91	99.02
100,000	1.99	9.99	99.90



Complejidad

- En general, las aplicaciones paralelas son mucho más complejas que las correspondientes aplicaciones en serie. No sólo porque tiene múltiples flujos de instrucciones que se ejecutan al mismo tiempo, sino que también hay datos que fluye entre ellos.
- Los costos de la complejidad se miden en tiempo de programación, en prácticamente todos los aspectos del ciclo de desarrollo de software:
 - ✓ Diseño
 - ✓ Codificación
 - ✓ Depuración
 - ✓ Pruebas
 - ✓ Mantenimiento
- La adhesión a las "buenas" prácticas de desarrollo de software, es esencial cuando se trabaja en aplicaciones paralelas - especialmente si alguien más tendrá que trabajar con el software.

Modelos de Programación Paralela

Información general

- Hay varios modelos de programación paralela de uso común:
 - ✓ De memoria compartida (sin hilos)
 - ✓ Threads (de hilos)
 - ✓ Memoria Distribuida / Paso de Mensajes
 - ✓ Datos Paralelos
 - ✓ Híbrido
 - ✓ Programa Individual de Datos Múltiples (SPMD)
 - ✓ Programa Múltiple de Datos Múltiples (MPMD)

¿**Qué modelo de usar?** Esto es a menudo una combinación de lo que está disponible y la elección personal. No existe el "mejor" modelo, aunque ciertamente hay mejores implementaciones de algunos modelos sobre otros.

Modelo Seleccionado para el Sistema

Modelo Threads (hilos)

- Este modelo de programación es un **tipo de programación de memoria compartida**.
- En el modelo de programación paralela, un proceso único "peso pesado" puede tener múltiples procesos de "peso ligero", rutas de ejecución concurrentes.
- Por ejemplo:
 - El programa principal a.out está programado para ejecutarse en el sistema operativo nativo. El a.out carga y adquiere todos los recursos de usuario necesarios para su ejecución. Este es el proceso de "peso pesado".
 - El a.out lleva a cabo algún trabajo en serie, y luego crea una serie de tareas (hilos) que se pueden programar y ejecutar en el sistema operativo al mismo tiempo.
 - Cada "hilo" (thread) no solo tiene datos locales, sino también, comparte la totalidad de los recursos del a.out. Esto ahorra la sobrecarga asociada a la replicación de los recursos de un programa para cada hilo ("peso ligero"). Cada hilo también se beneficia de una vista de la memoria global, ya que comparte el espacio de memoria del a.out.

- El trabajo de un hilo puede ser mejor descrito como una subrutina dentro del programa principal. Cualquier hilo puede ejecutar cualquier subrutina al mismo tiempo que otros hilos.
- Los hilos se comunican entre sí a través de la memoria global (actualización de ubicaciones de direcciones). Esto requiere de construcciones de sincronización para asegurarse de que más de un hilo no está actualizando la misma dirección global en cualquier momento.
- Los hilos pueden ir y venir, pero el a.out permanece presente para proporcionar los recursos compartidos necesarios hasta que la aplicación se ha completado.

4.12 MAPAS DE PELIGRO

Para la zona de estudio se elaboraron los mapas de peligro con apoyo de un sistema de información geográfica con base en los resultados de las modelaciones matemáticas de flujos de inundaciones referentes a los valores máximos (envolventes) de P_m , V_m y $(PV^2)_m$ para los periodos de retorno de 2, 5, 10, 50 y 100 años. En ellos, aparecerá el tiempo de permanencia de las inundaciones. Se establecen algunas conclusiones sobre el trabajo realizado para formar estos mapas y los resultados obtenidos. A continuación se presenta una imagen de los mapas.

4.12.1 Mapas de severidad

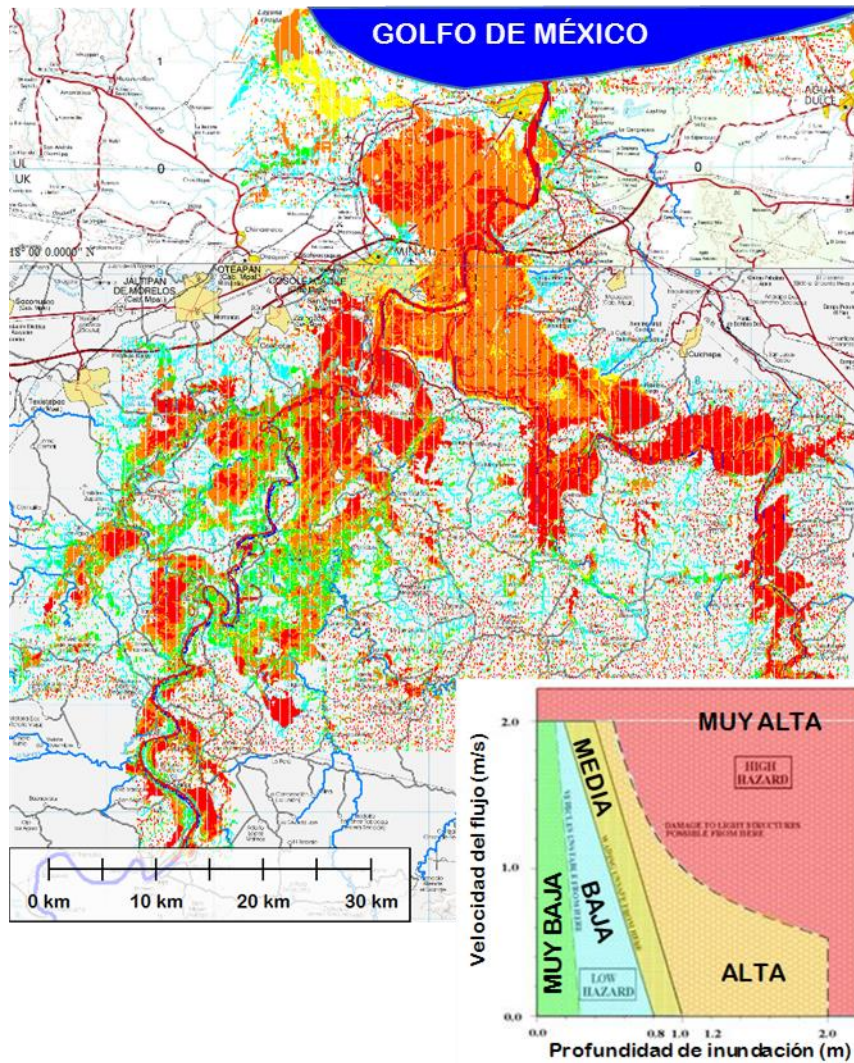


Figura 12.1 Mapa de severidad para Tr 2

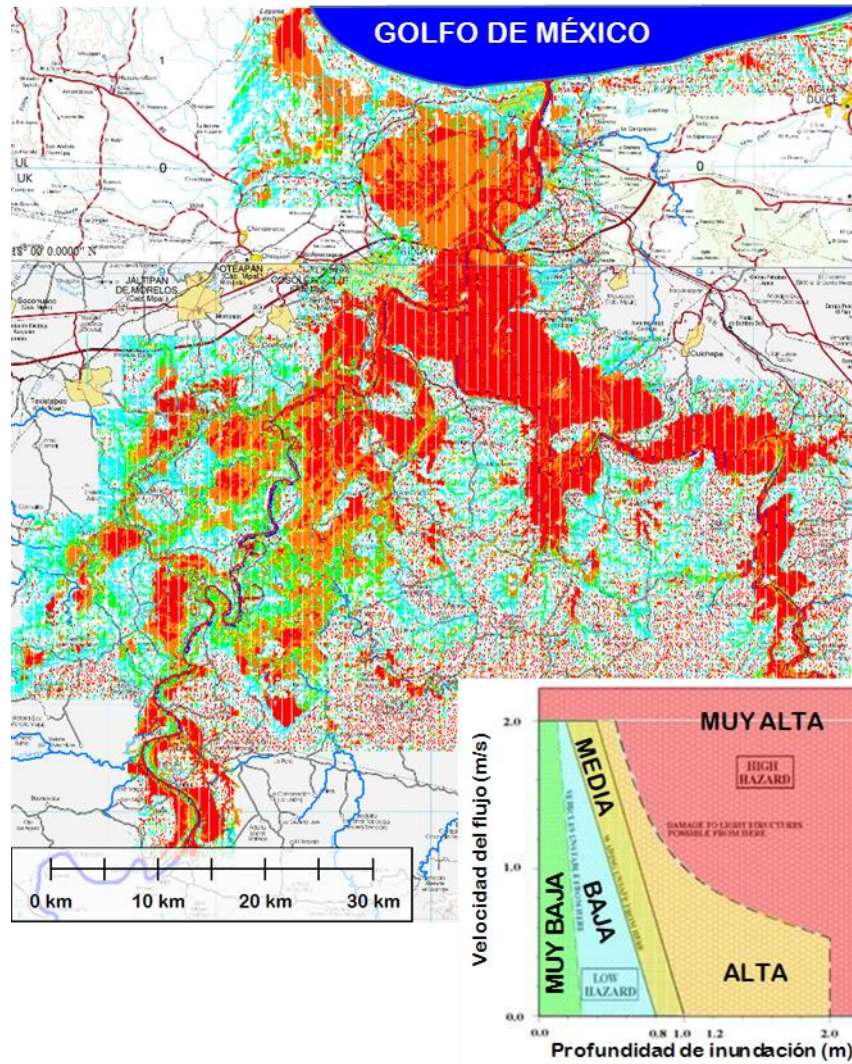


Figura 12.2 Mapa de severidad para Tr 5

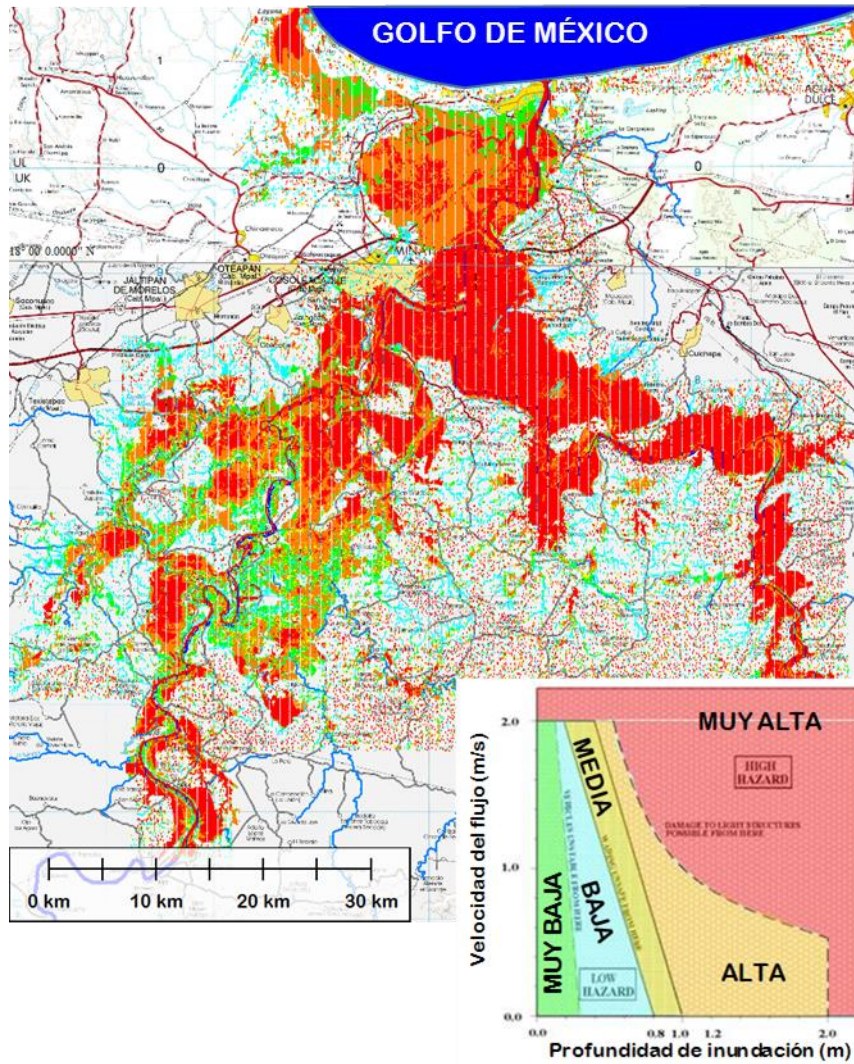


Figura 12.3 Mapa de severidad para Tr 10

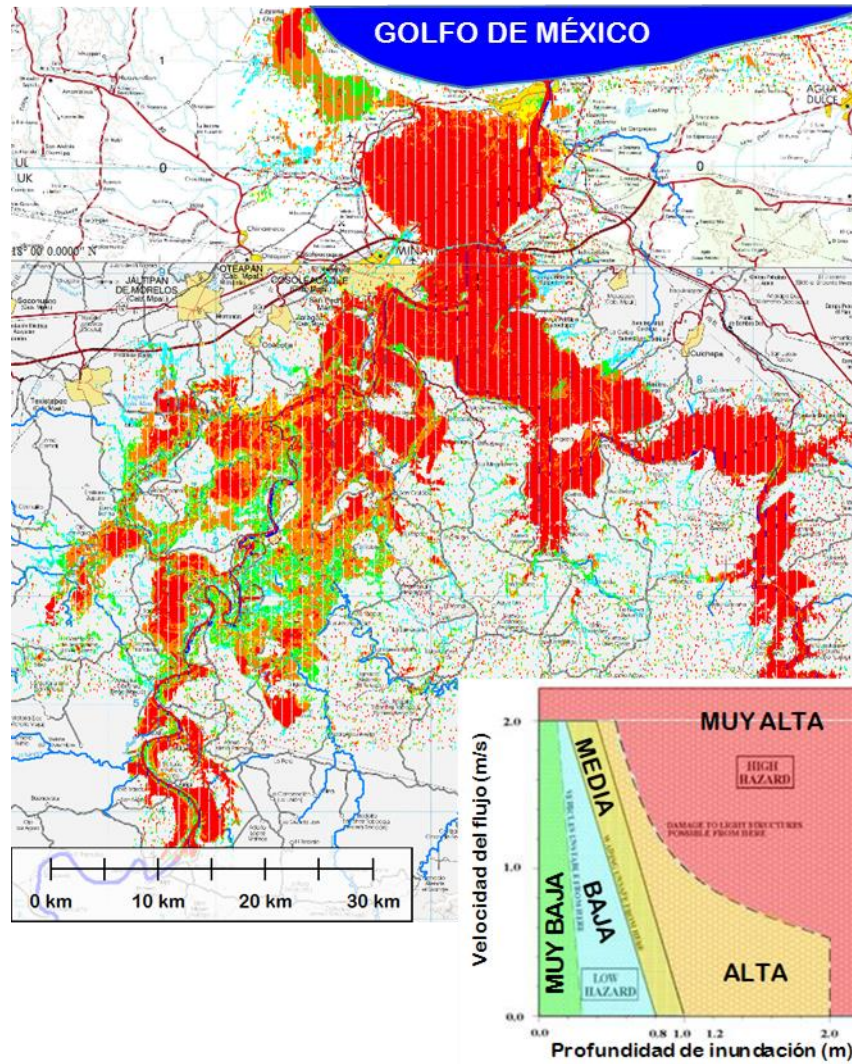


Figura 12.4 Mapa de severidad para Tr 50

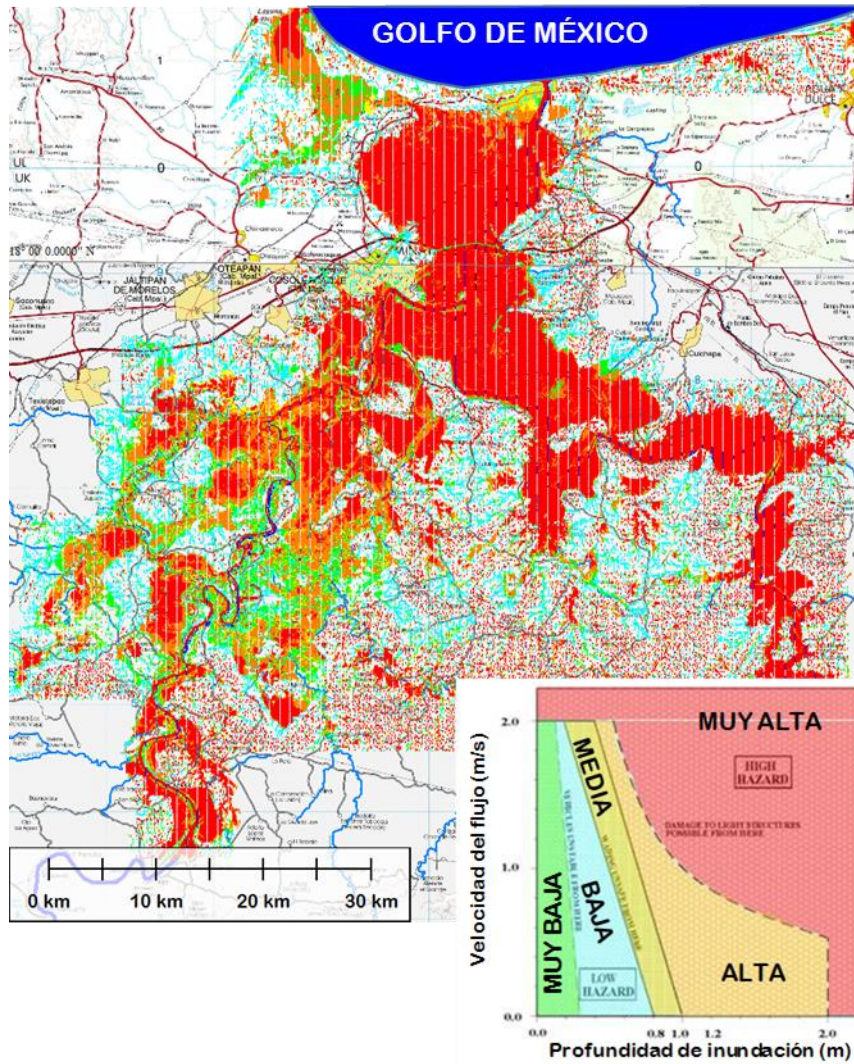


Figura 12.5 Mapa de severidad para Tr 100

4.12.2 Mapas de profundidades máximas

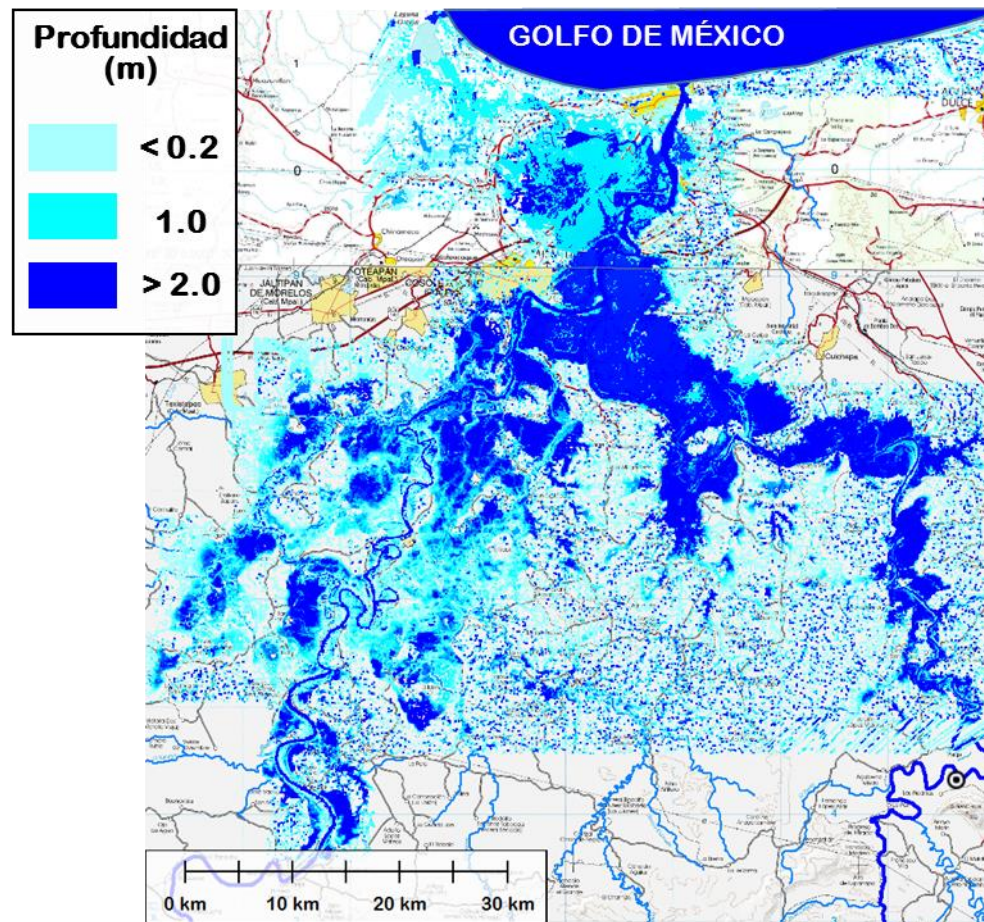


Figura 12.6 Mapa de profundidades máximas para Tr 2

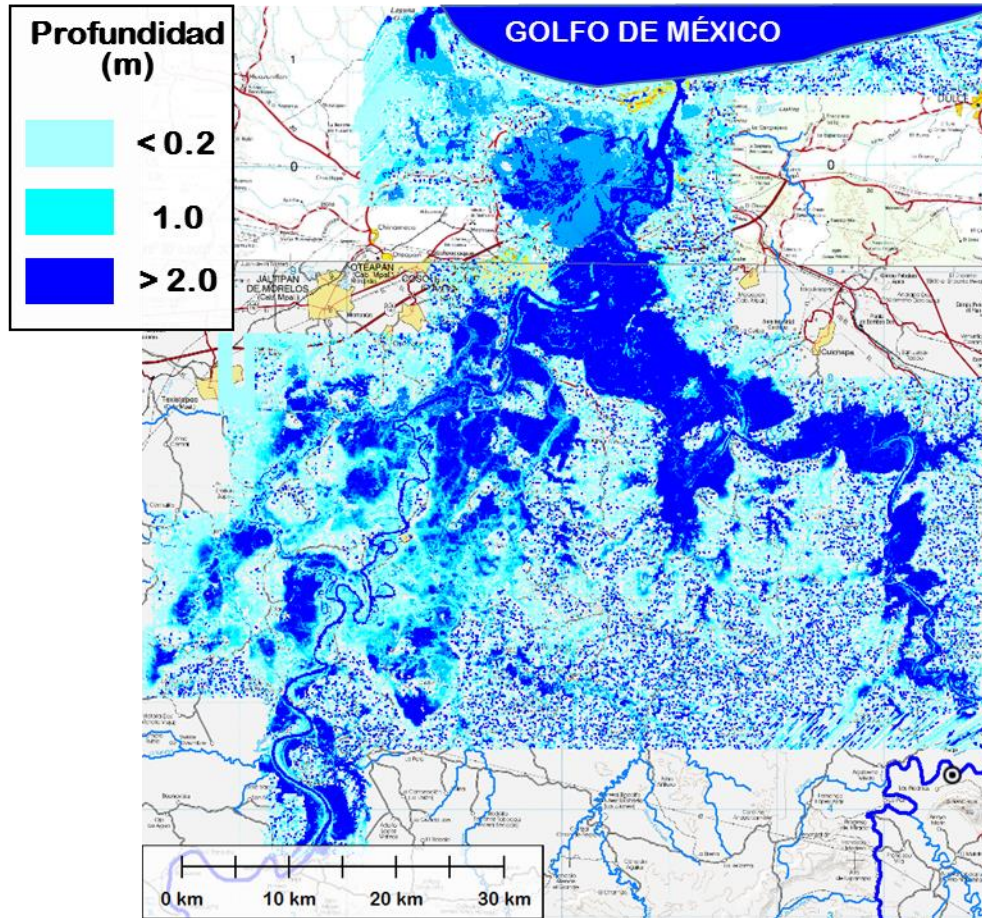


Figura 12.7 Mapa de profundidades máximas para Tr 5

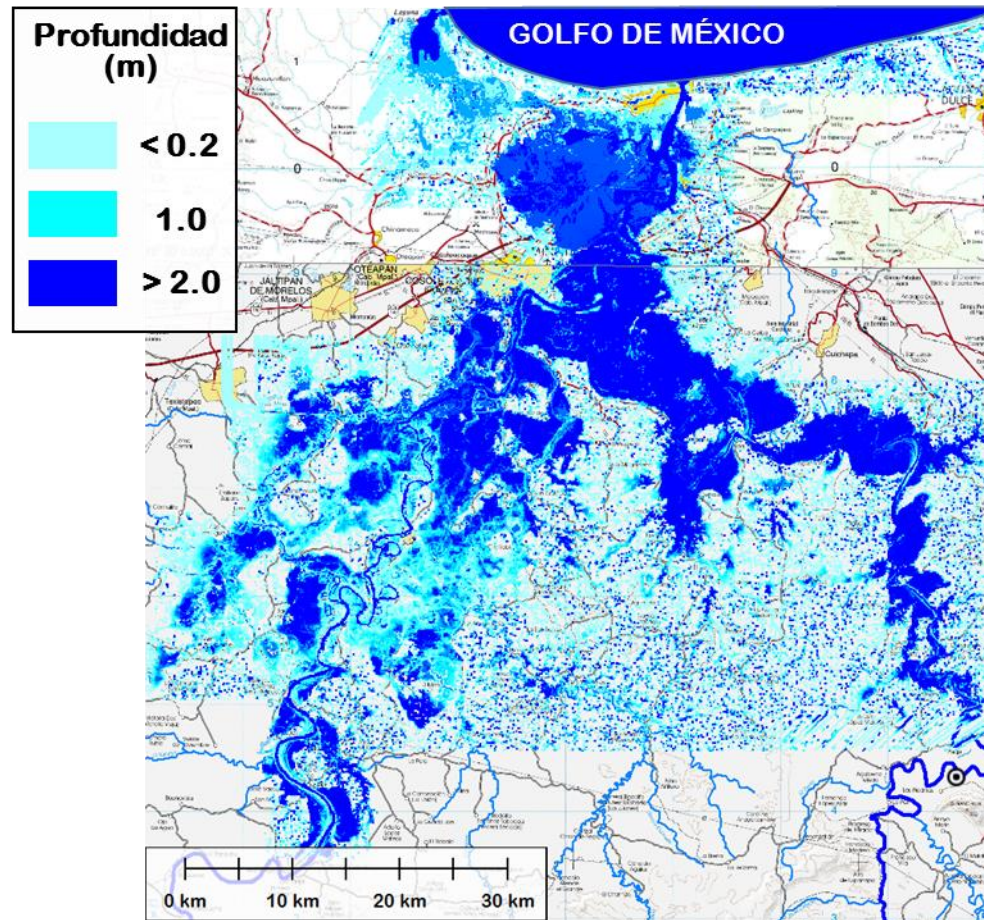


Figura 12.8 Mapa de profundidades máximas para Tr 10

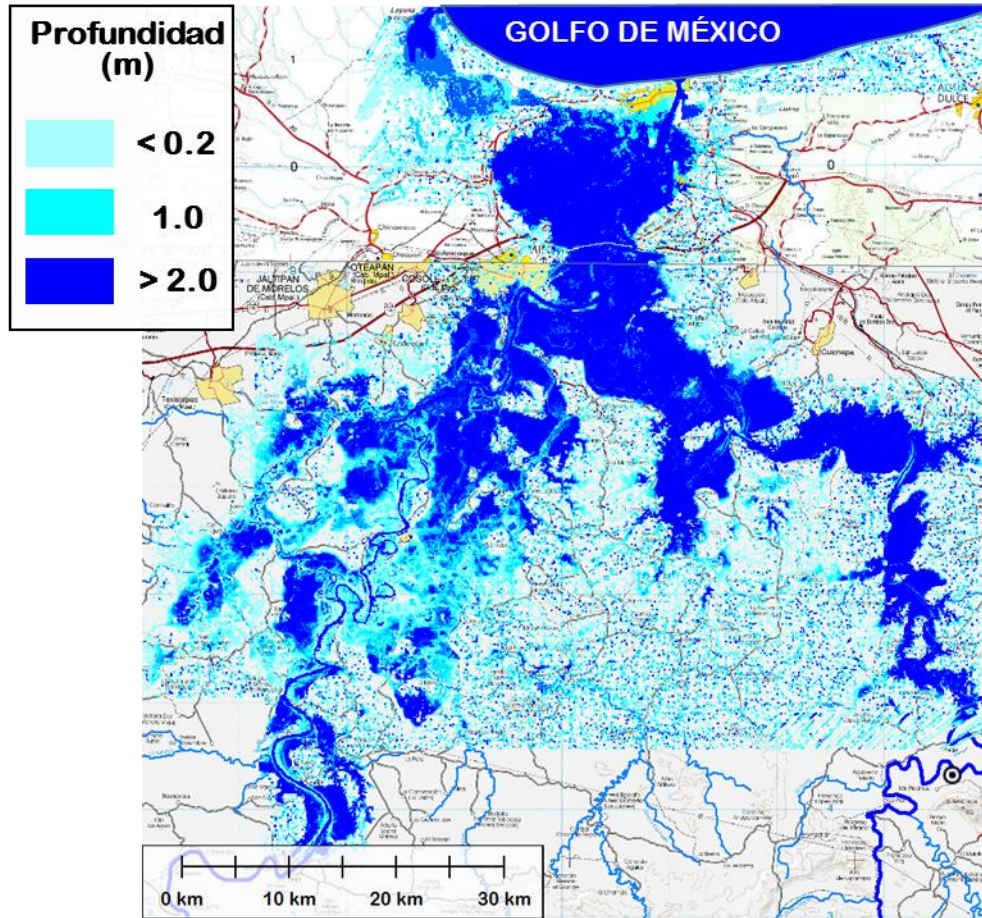


Figura 12.9 Mapa de profundidades máximas para Tr 50

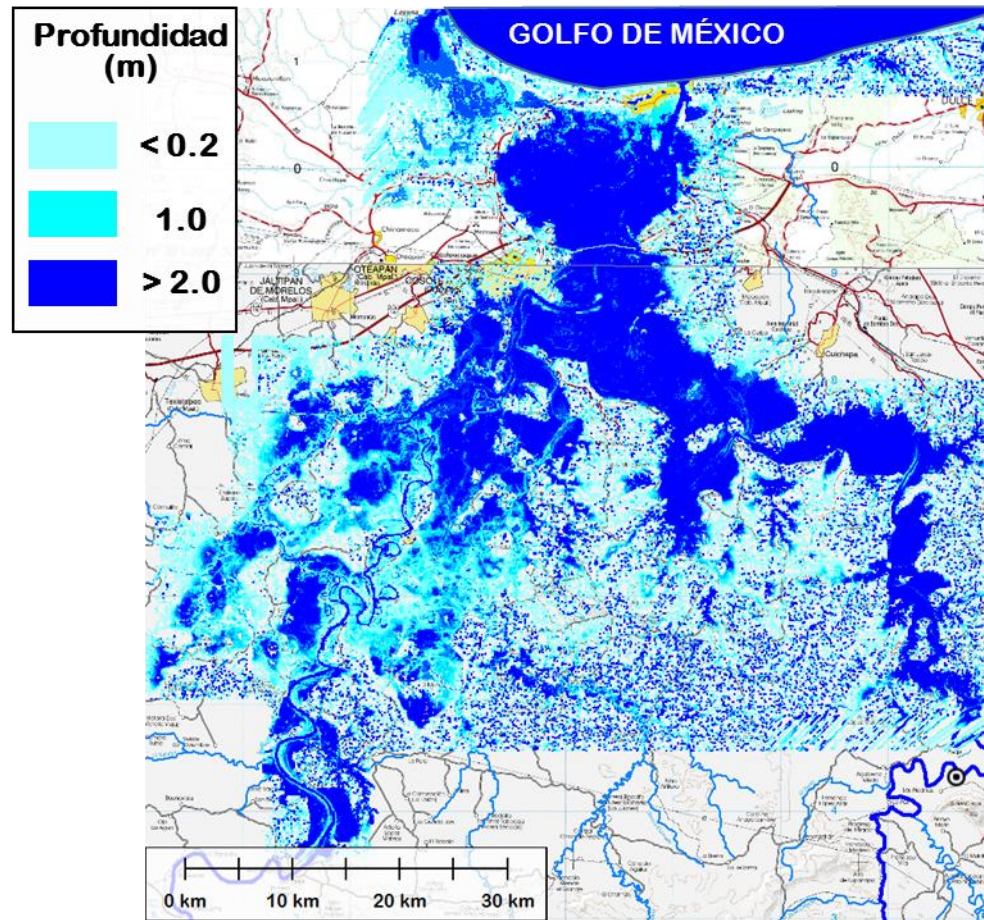


Figura 12.10 Mapa de profundidades máximas para Tr 100

4.12.3 Mapas de velocidades máximas

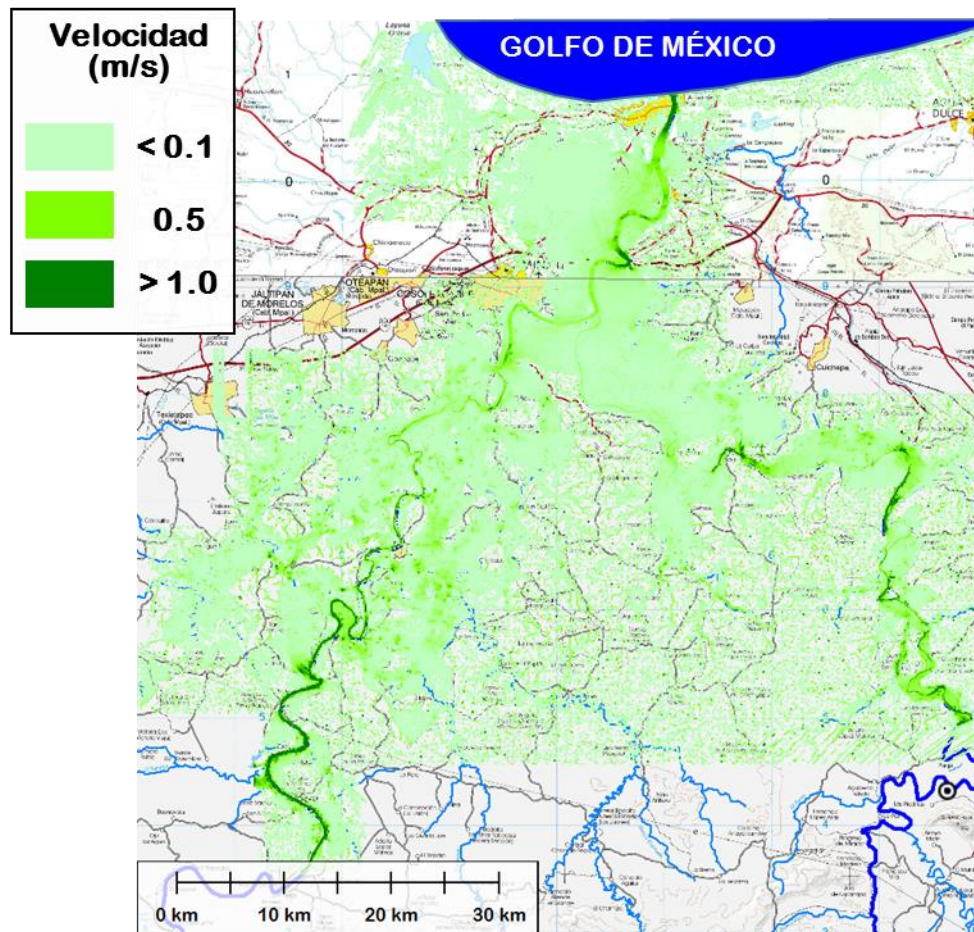


Figura 12.11 Mapa de velocidades máximas para Tr 2

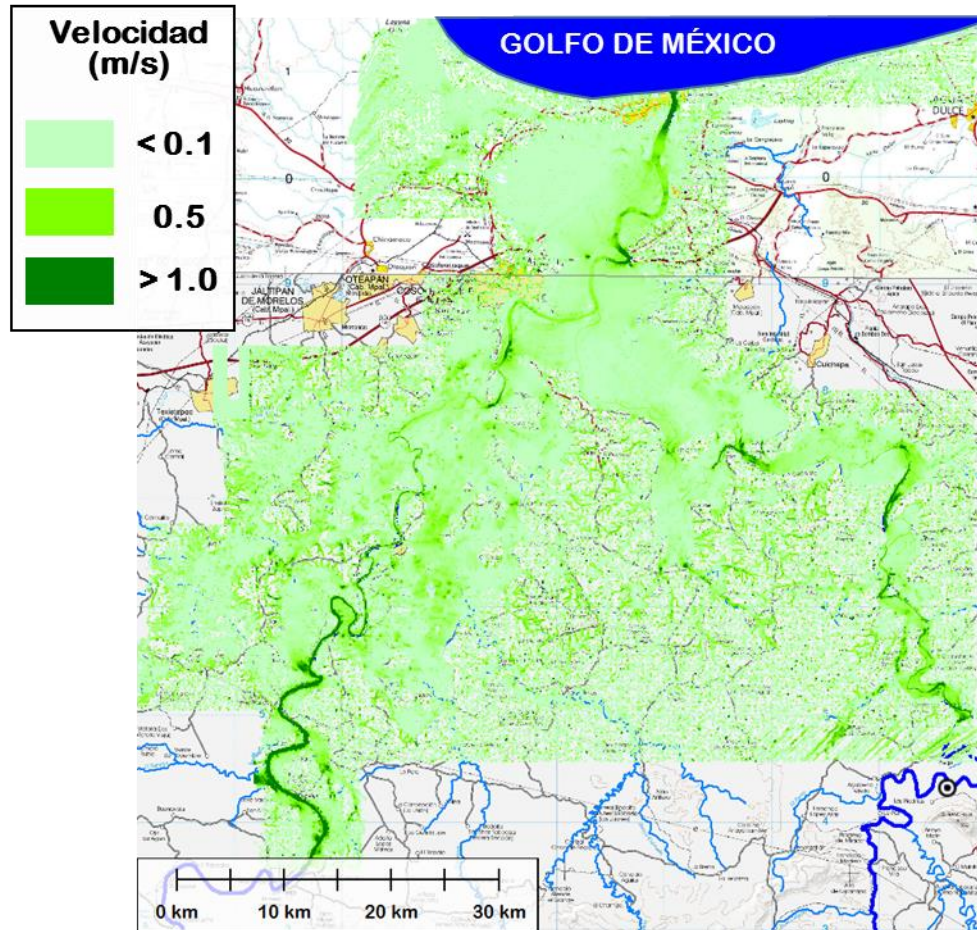


Figura 12.12 Mapa de velocidades máximas para Tr 5

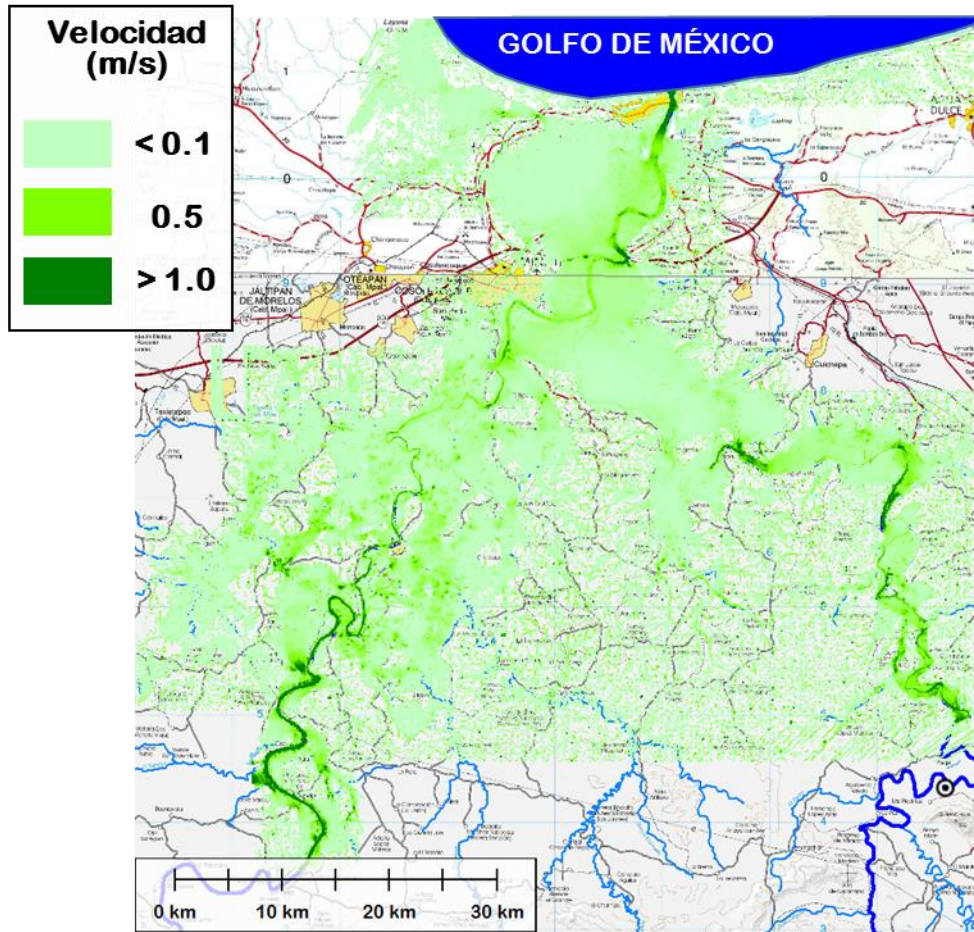


Figura 12.13 Mapa de velocidades máximas para Tr 10

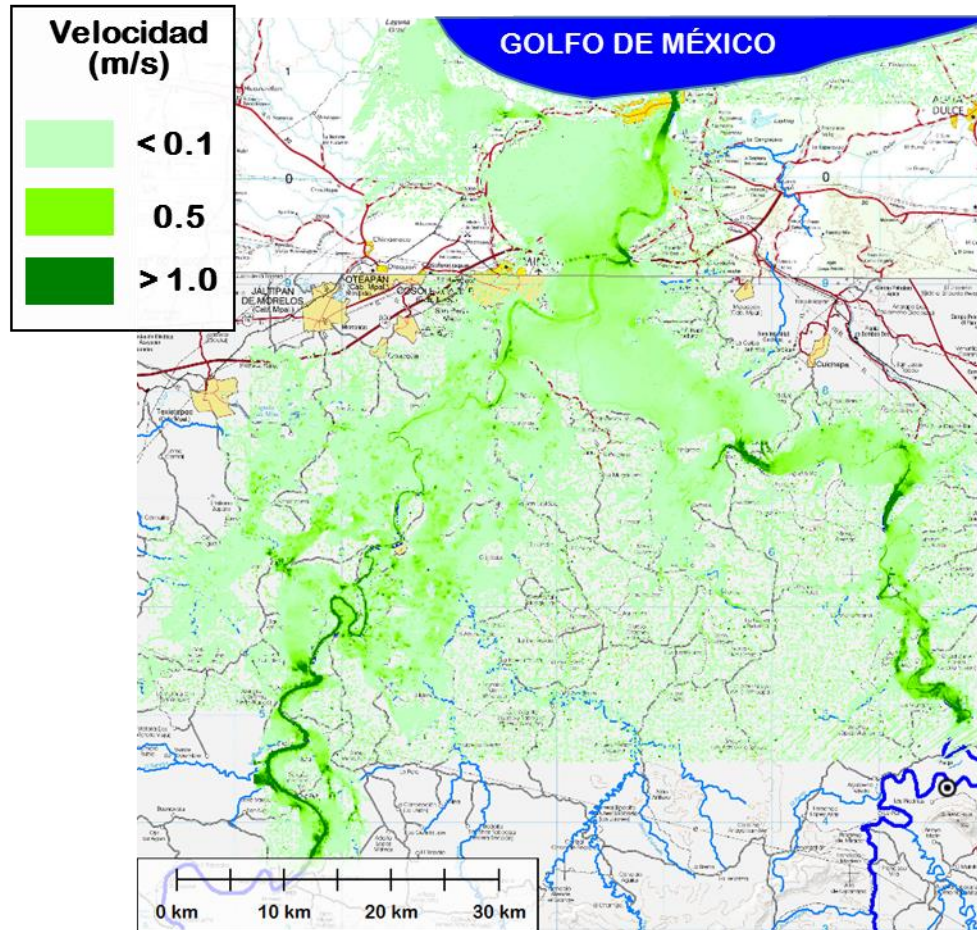


Figura 12.14 Mapa de velocidades máximas para Tr 50

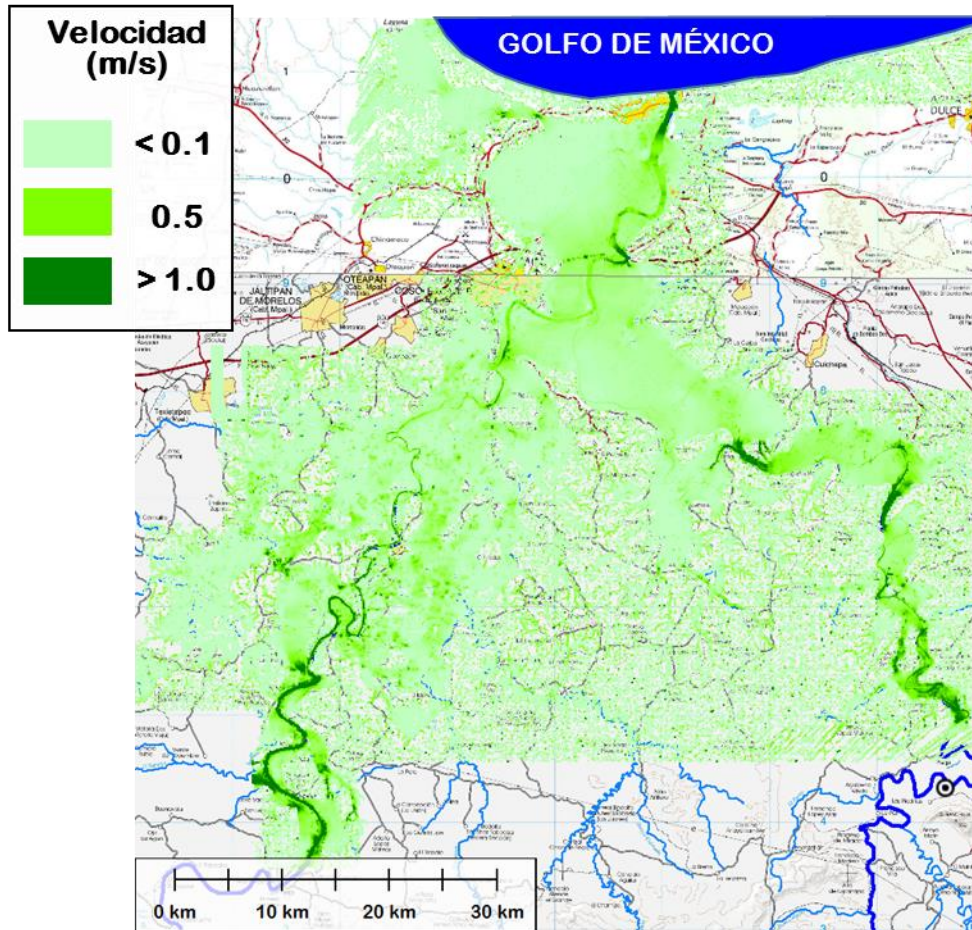


Figura 12.15 Mapa de velocidades máximas para Tr 100

Zapata, C. J. (2005), Licuación de arenas en la refinería General Lázaro Cárdenas, Tesis presentada a la División de Estudios de Posgrado de la Facultad de Ingeniería, UNAM, para obtener el grado de Maestro en Ingeniería.

REFERENCIAS BIBLIOGRAFICAS DE LOS CAPITULOS 4.3 AL 4.12

Alvarado, C.O. Estudio de mercado para un proyecto, presentado para los estudiantes de arquitectura de la UCA, Nicaragua, Marzo 12 de 2009.

BAEZA, R. C. (2007). "Estimación regional de factores de convectividad para el cálculo de las relaciones intensidad-duración-frecuencia". Tesis de Maestría. Universidad Nacional Autónoma de México.

BARÓ, Suarez José Emilio. "Costo más probable de daños por inundación en zonas habitacionales de México", septiembre 2011.

CHEN, C. L. (1983). Rainfall intensity-duration-frequency formulas. Journal of Hydraulic Engineering, ASCE, Vol. 109, No. 12, December 1983, pp. 1603-1621.

CHOW, V. T. (1994). "Hidrología aplicada". McGraw-Hill Interamericana, S.A., Colombia.

FUENTES, M. O. A. (2012). "Obtención de hietogramas correspondientes a diferentes periodos de retorno", XXV Congreso Latinoamericano de Hidráulica, San José, Costa Rica.

OMM-No.1047 Gestión integrada de crecidas: Documento Conceptual. Organización Meteorológica Mundial.. Ginebra, Suiza 2009.

OMM, Tercera Conferencia Mundial sobre el clima, Ginebra, Suiza, 31 de agosto-4 de septiembre de 2009.

Paoli, C.U. Curso Gestión Integrada de Crecidas, Facultad de Ingeniería y Ciencias Hídricas- Universidad Nacional del Litoral. Centro Regional Litoral-Instituto Nacional del Agua.-Prof., Santa Fe Argentina, 26 al 30 de abril de 2010.

Salas M. A. Metodología para la elaboración de mapas de riesgo por inundaciones en zonas urbanas. Centro Nacional de Prevención de Desastres. Serie: Atlas Nacional de Riesgos. Fenómenos Hidrometeorológicos., Julio 2011.